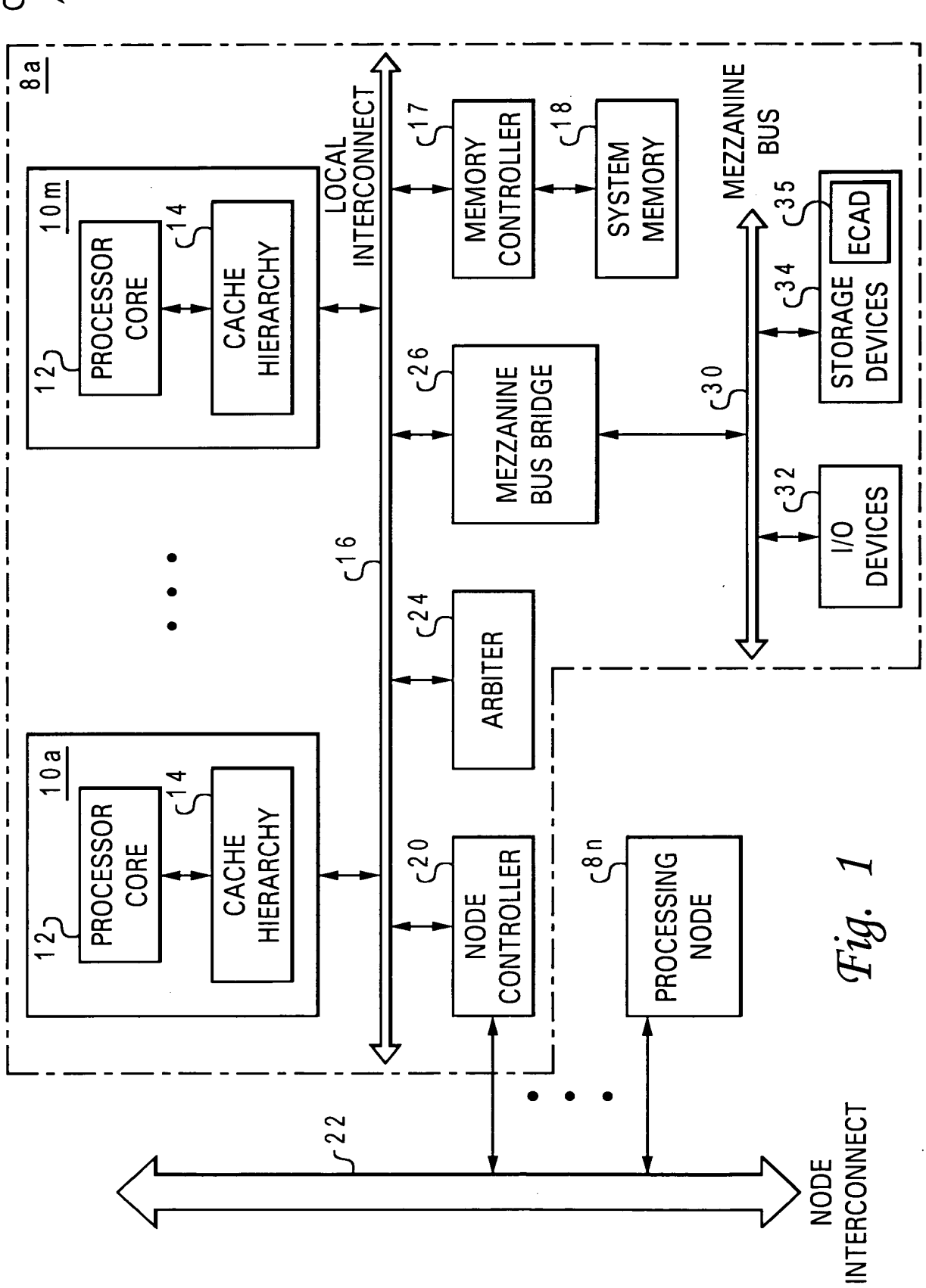

Fig. 1

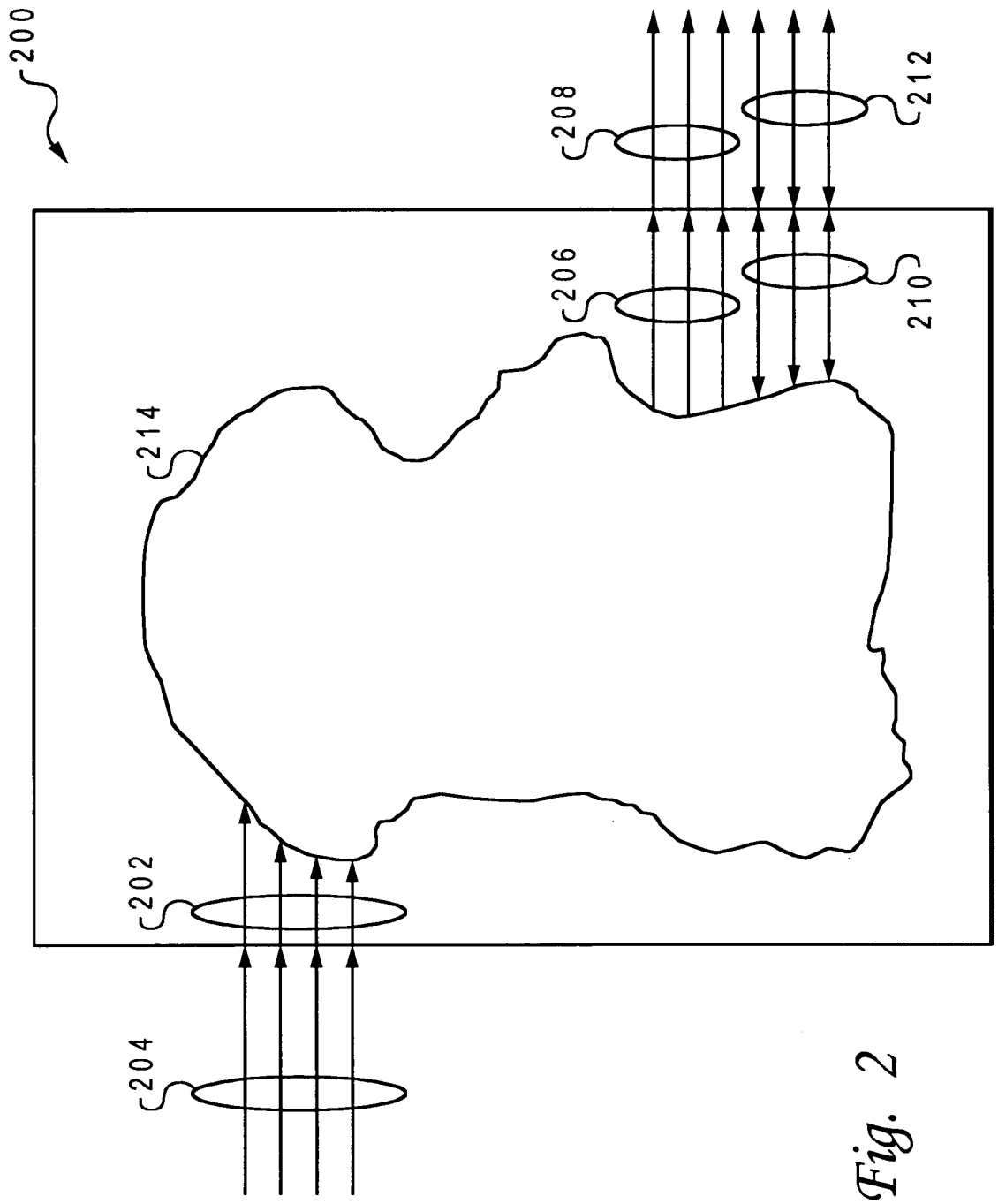

Fig. 2

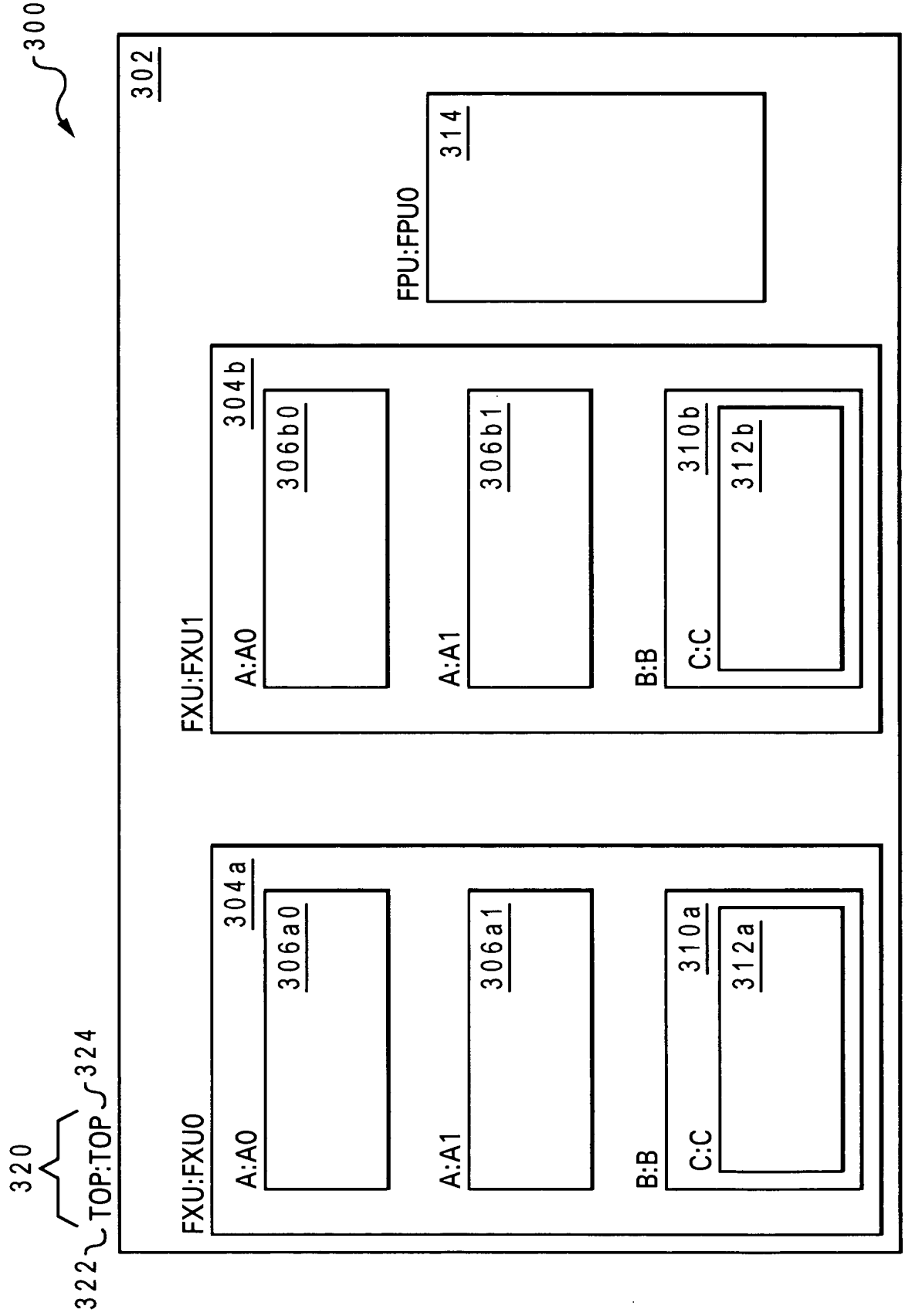
300
302
FXU:FXU0
A:A0
304a
306a0
A:A1
306a1
B:B
C:C
310a
312a
FXU:FXU1
A:A0
304b
306b0
A:A1
306b1
B:B
C:C
310b
312b
FPU:FPU0
314
320
322
TOP:TOP
324

Fig. 3

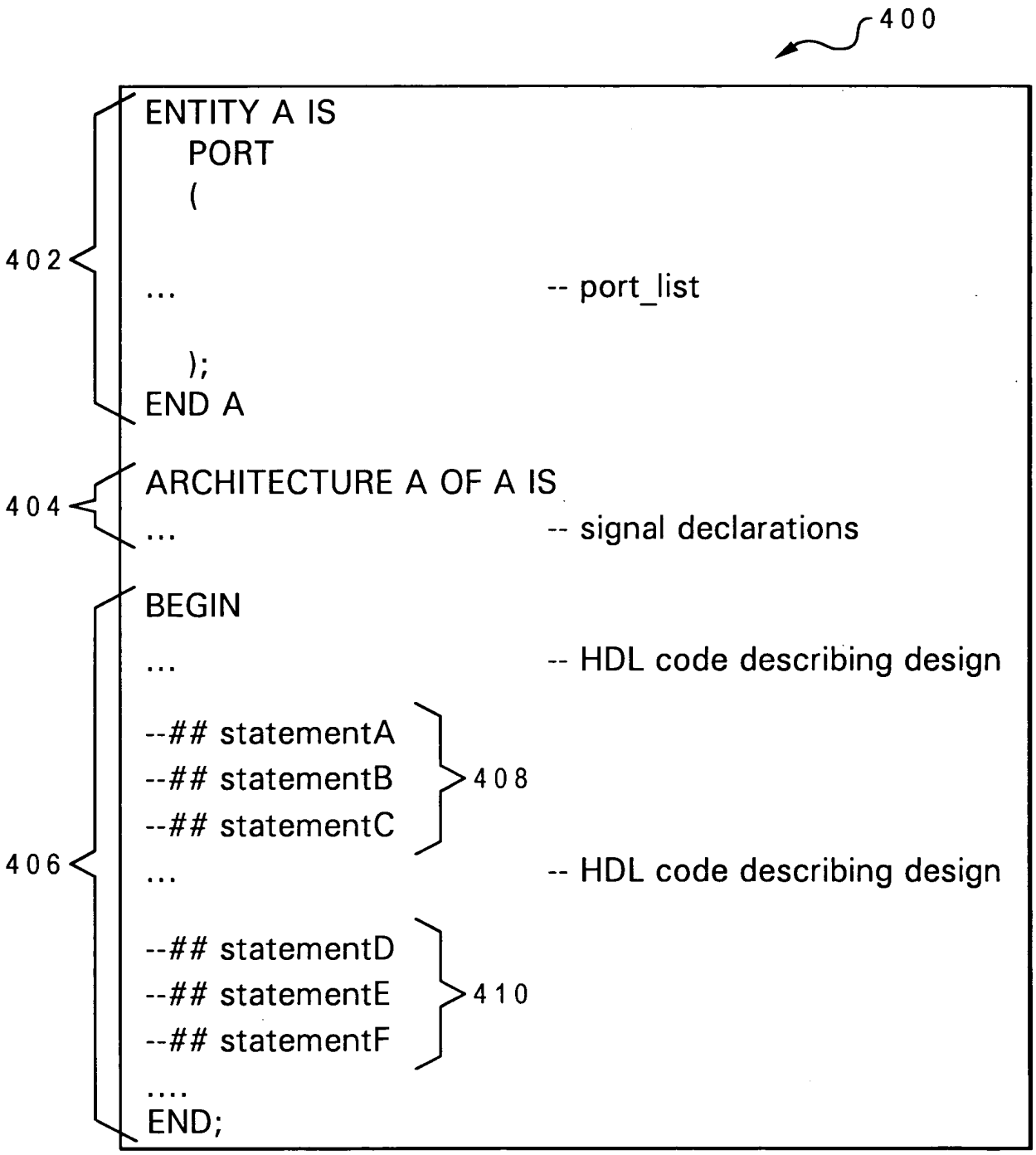
400
ENTITY A IS
PORT
(
...
-- port_list
);
END A
ARCHITECTURE A OF A IS
...
-- signal declarations
BEGIN
...
-- HDL code describing design
--## statementA
--## statementB
--## statementC
...
-- HDL code describing design
--## statementD
--## statementE
--## statementF
....
END;
402
404
406
408
410

Fig. 4A

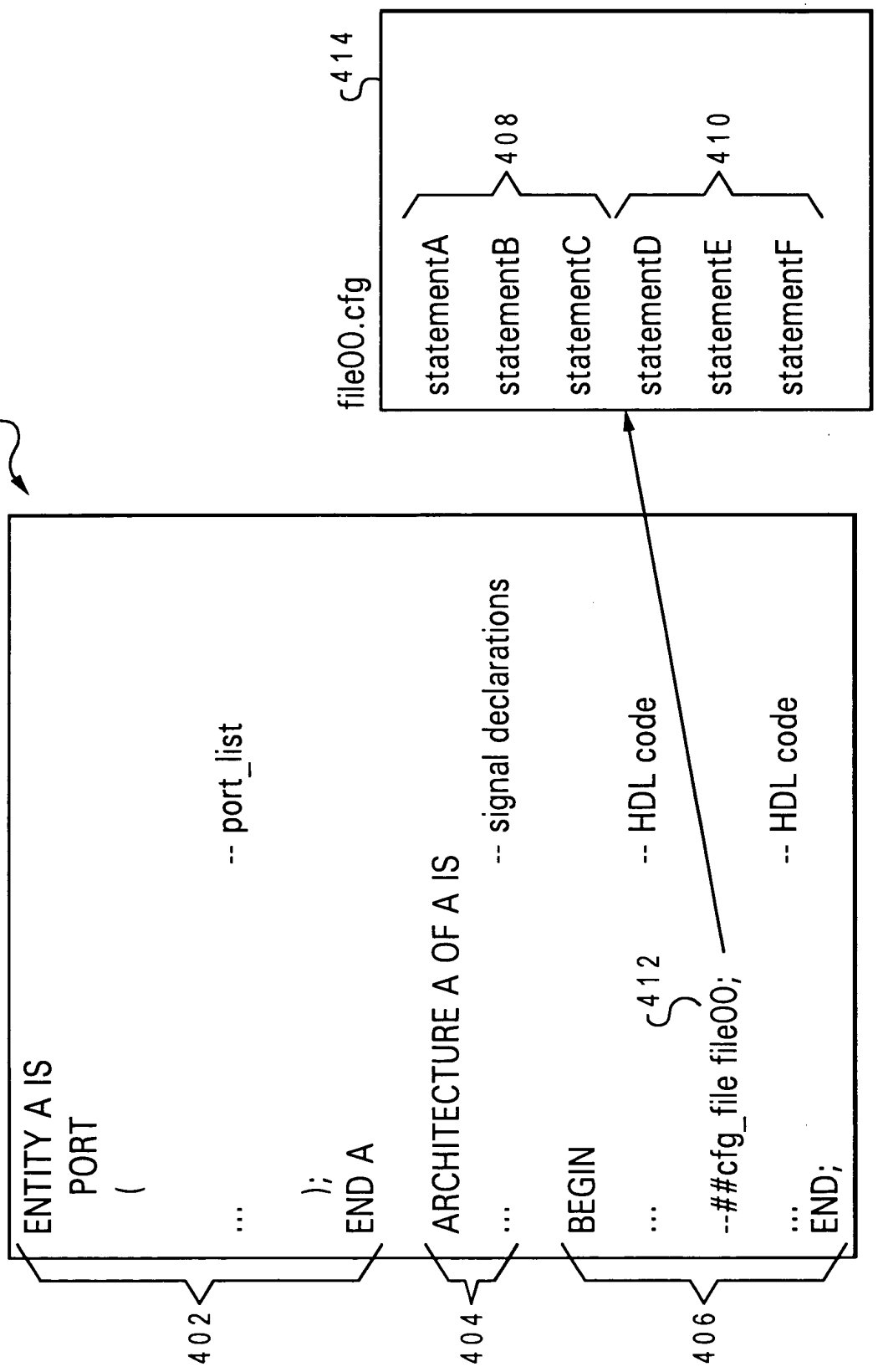
400'
ENTITY A IS
PORT
(
...
-- port_list
);
END A
ARCHITECTURE A OF A IS
...
-- signal declarations
BEGIN
...
-- HDL code
412
--##cfg_file file00;
...
-- HDL code
END;
402
404
406
414
file00.cfg
statementA
statementB
statementC
statementD
statementE
statementF
408
410

Fig. 4B

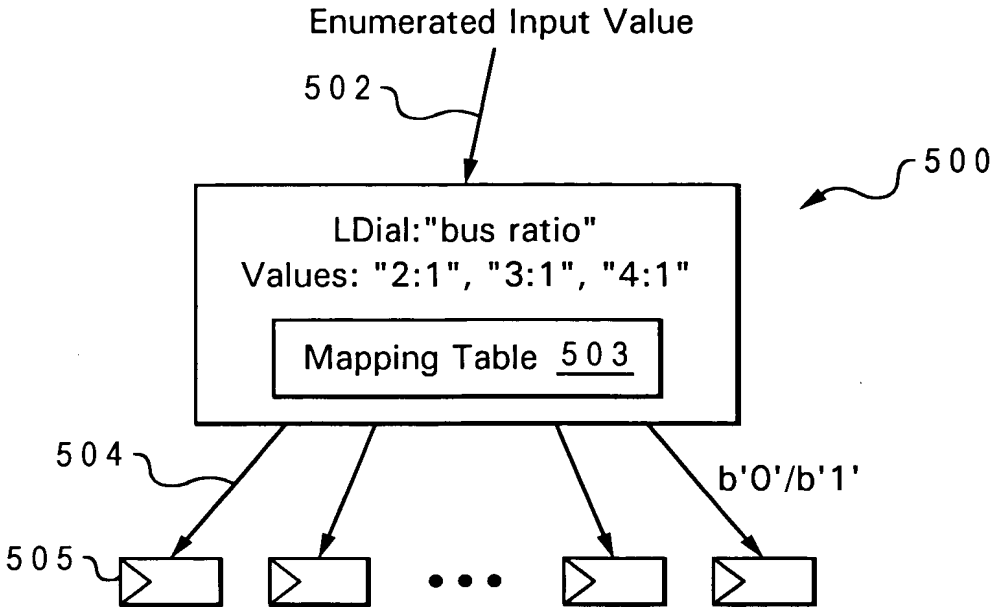
Enumerated Input Value
502
500
LDial:"bus ratio"
Values: "2:1", "3:1", "4:1"
Mapping Table 503
504
b'0'/b'1'
505

Fig. 5A

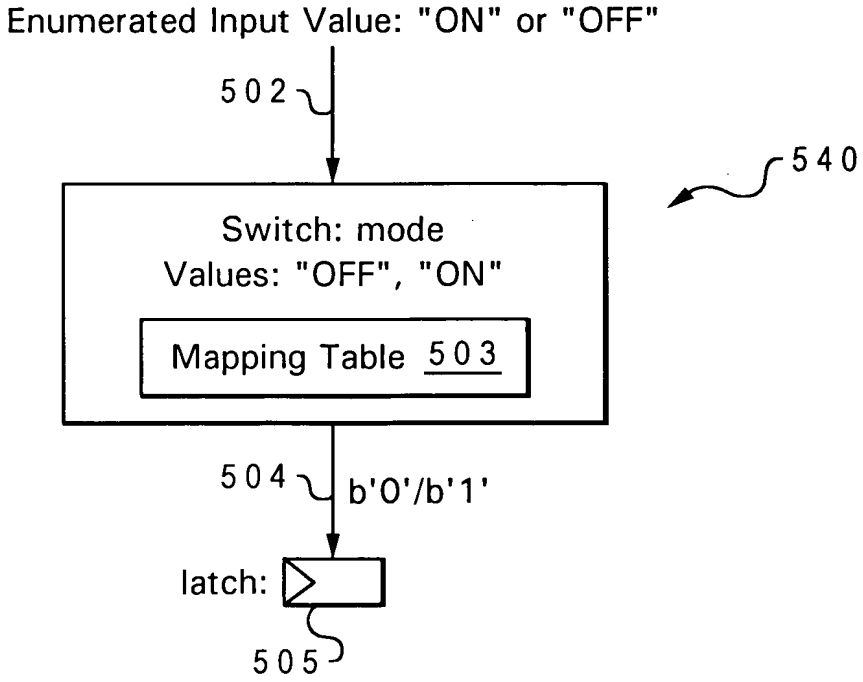
Enumerated Input Value: "ON" or "OFF"
502
540
Switch: mode
Values: "OFF", "ON"
Mapping Table 503
504
b'0'/b'1'
latch:
505

Fig. 5D

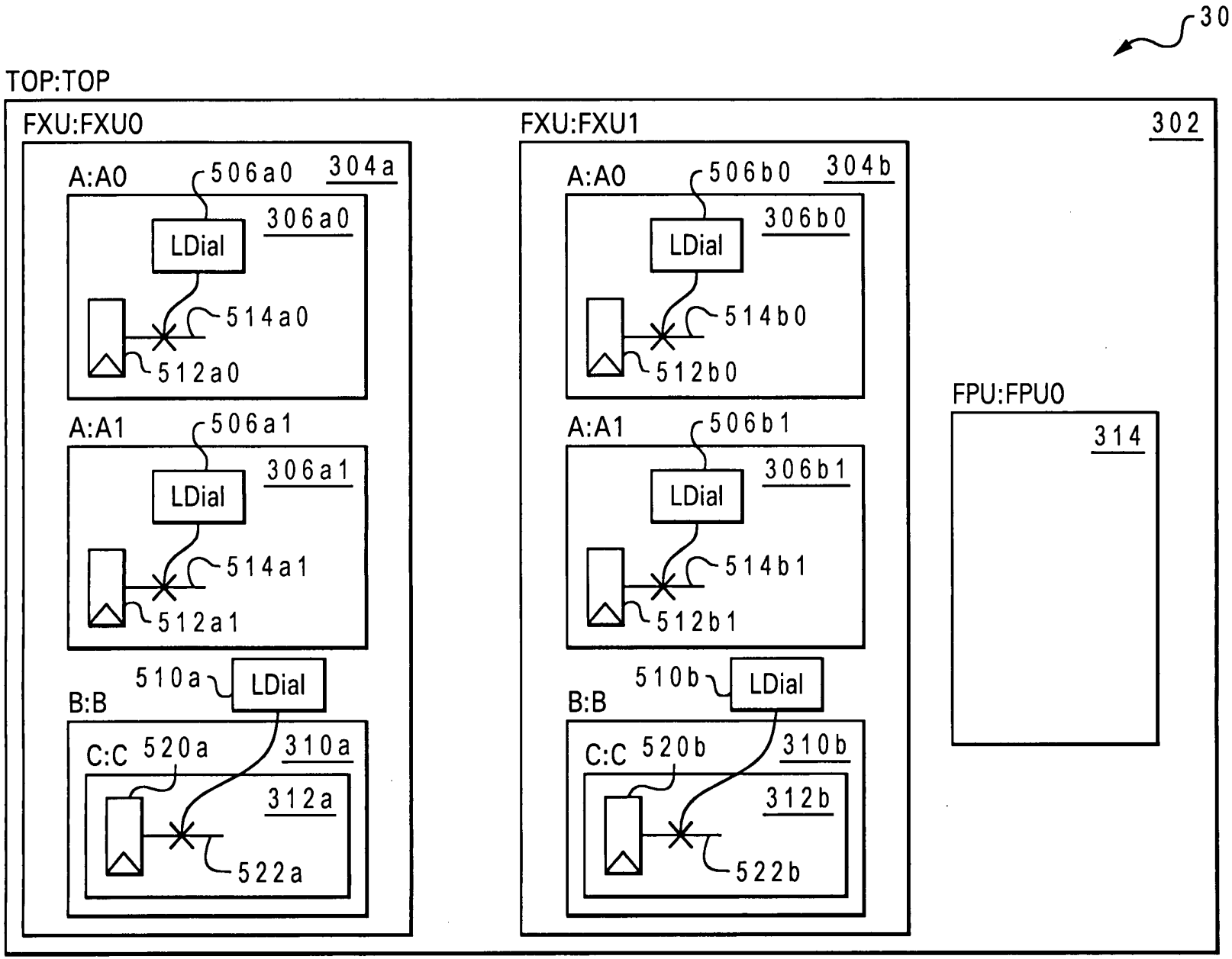
300'
TOP:TOP
FXU:FXU0
A:A0
506a0
304a
306a0
LDial
514a0
512a0
A:A1
506a1
306a1
LDial
514a1
512a1
510a
LDial
B:B
C:C
520a
310a
312a
522a
FXU:FXU1
302
A:A0
506b0
304b
306b0
LDial
514b0
512b0
A:A1
506b1
306b1
LDial
514b1
512b1
510b
LDial
B:B
C:C
520b
310b
312b
522b
FPU:FPU0
314

Fig. 5B

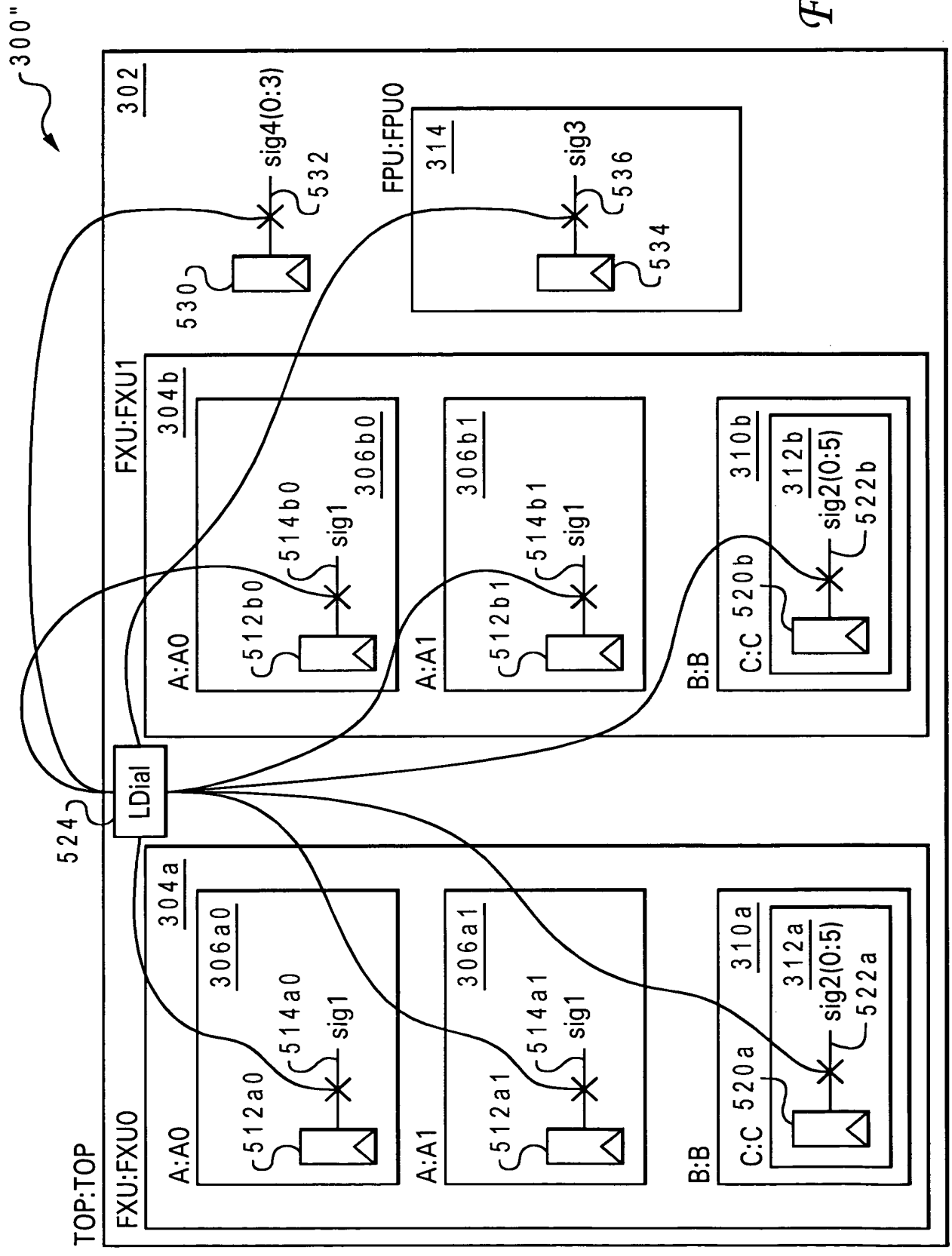
300"
302
TOP:TOP
FXU:FXU0
304a
A:A0
306a0
512a0
514a0
sig1
A:A1
306a1
512a1
514a1
sig1
B:B
310a
C:C 520a
312a
sig2(0:5)
522a
524
LDial
FXU:FXU1
304b
A:A0
306b0
512b0
514b0
sig1
A:A1
306b1
512b1
514b1
sig1
B:B
310b
C:C 520b
312b
sig2(0:5)
522b
530
sig4(0:3)
532
FPU:FPU0
314
sig3
536
534

Fig. 5C

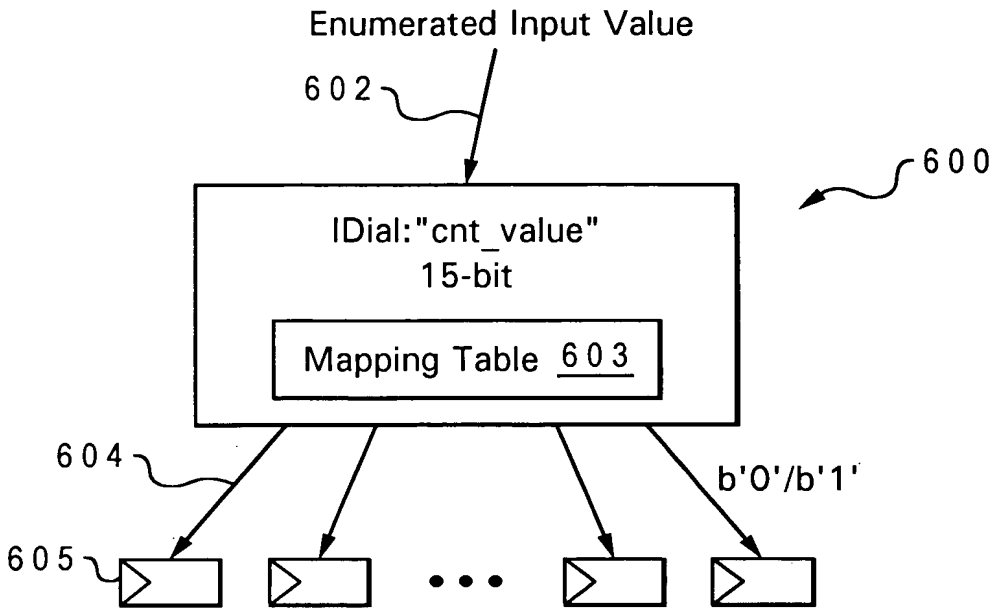
Enumerated Input Value
602
600
IDial:"cnt_value"
15-bit
Mapping Table 603
604
b'0'/b'1'
605

Fig. 6A

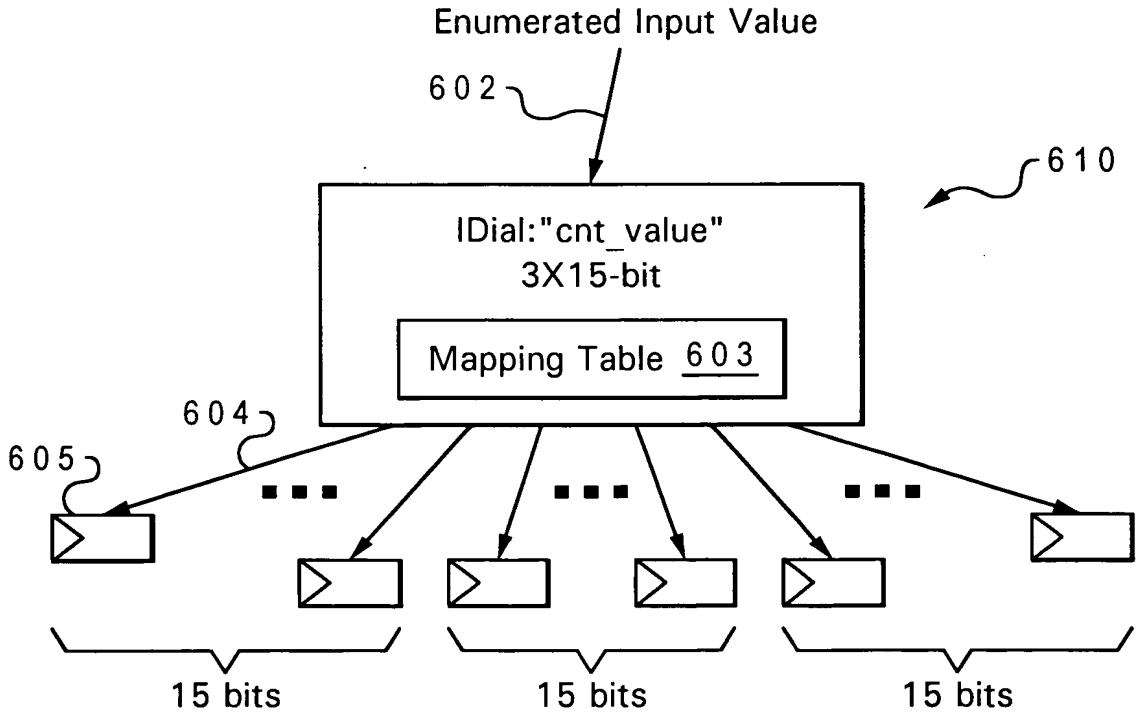
Enumerated Input Value
602
610
IDial:"cnt_value"
3X15-bit
Mapping Table 603
604
605
15 bits
15 bits
15 bits

Fig. 6B

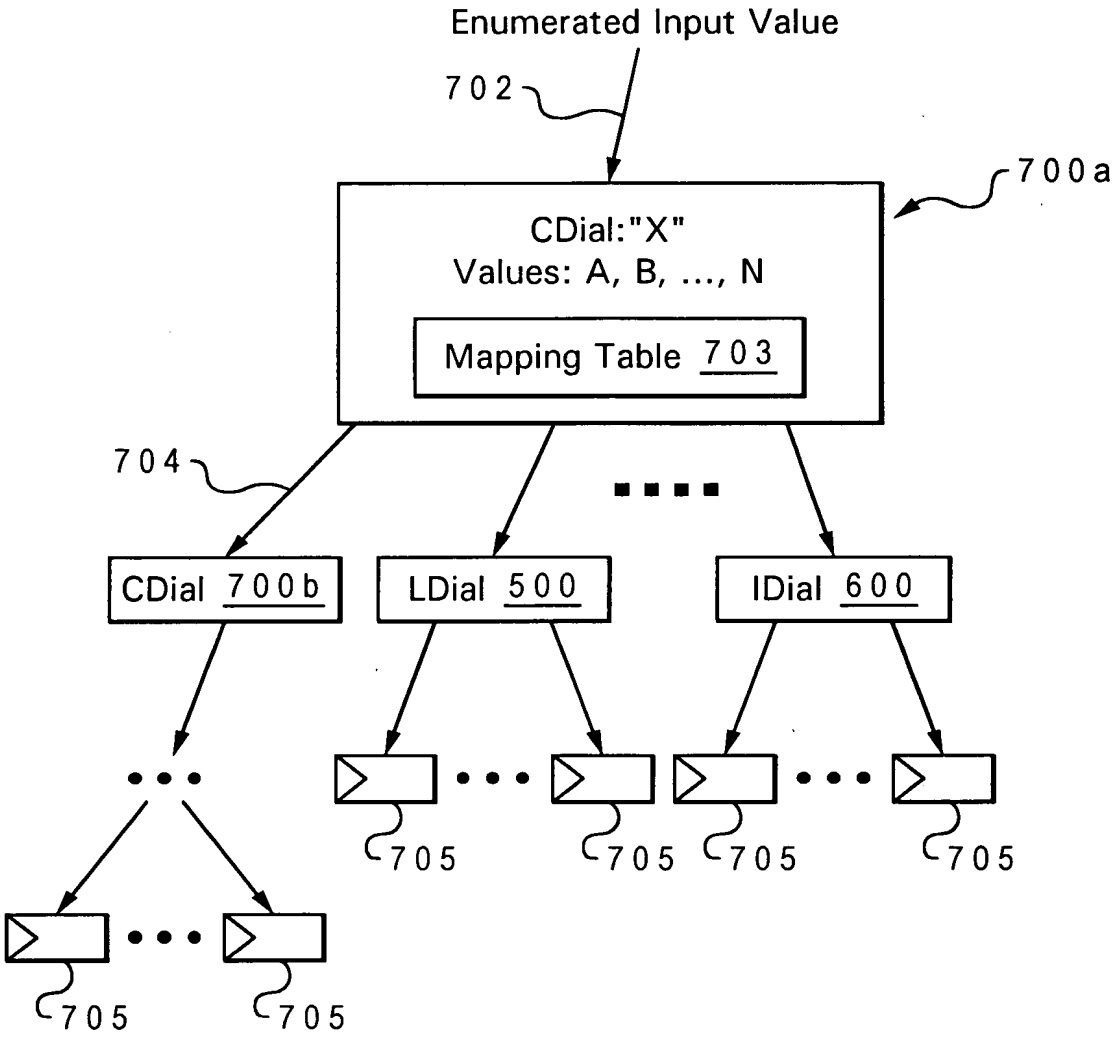
Enumerated Input Value
702
700a
CDial:"X"
Values: A, B, ..., N
Mapping Table 703
704
CDial 700b
LDial 500
IDial 600
705
705
705
705
705
705

Fig. 7A

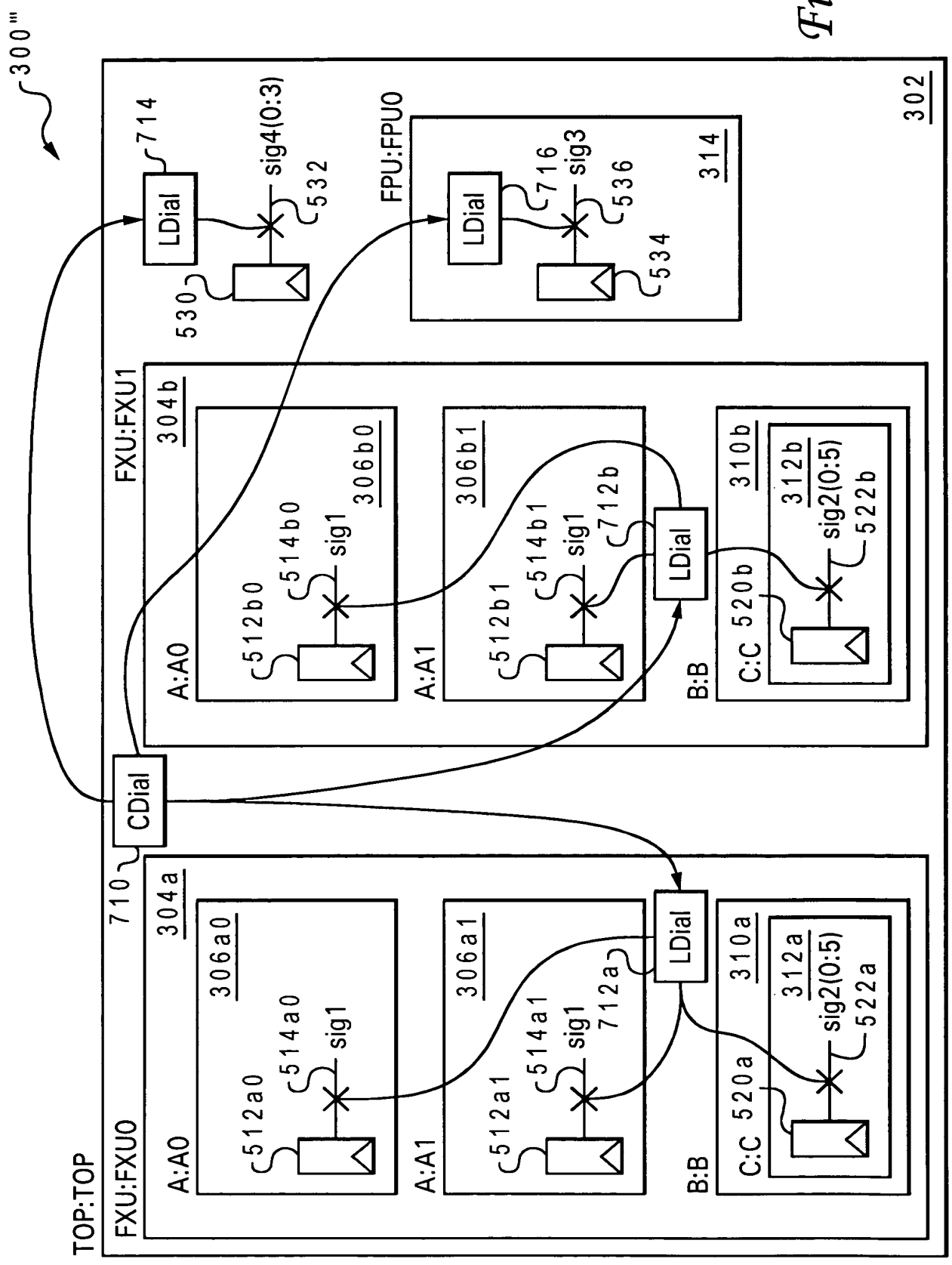
300"
TOP:TOP
FXU:FXU0
710
CDial
A:A0
304a
306a0
512a0
514a0
sig1
A:A1
306a1
512a1
514a1
712a
LDial
B:B
310a
C:C
520a
312a
sig2(0:5)
522a
302
FXU:FXU1
304b
306b0
512b0
514b0
306b1
512b1
514b1
712b
310b
520b
312b
522b
FPU:FPU0
314
714
LDial
sig4(0:3)
532
530
716
sig3
536
534

Fig. 7B

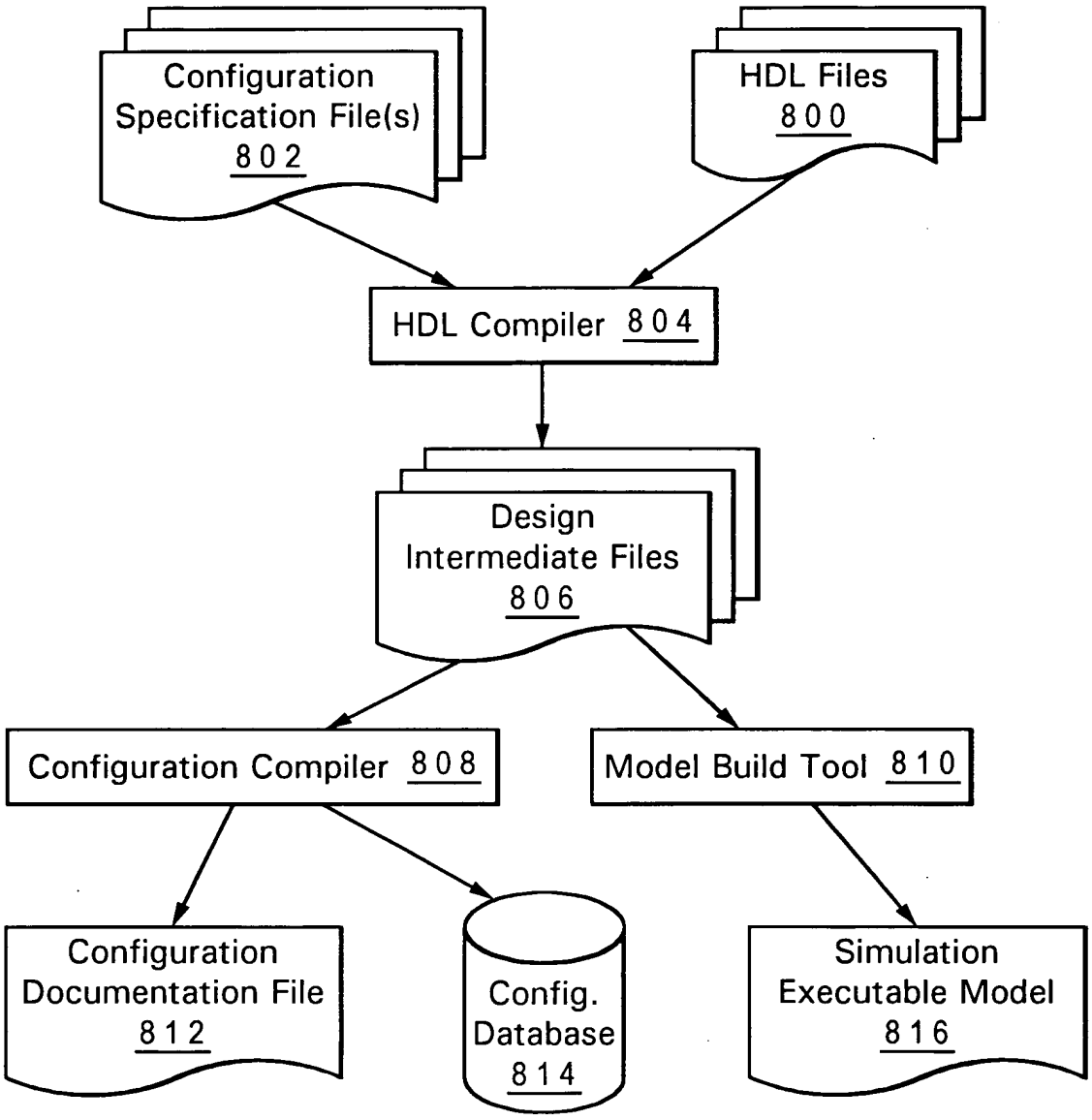
Configuration Specification File(s) 802
HDL Files 800
HDL Compiler 804
Design Intermediate Files 806
Configuration Compiler 808
Model Build Tool 810
Configuration Documentation File 812
Config. Database 814
Simulation Executable Model 816

Fig. 8

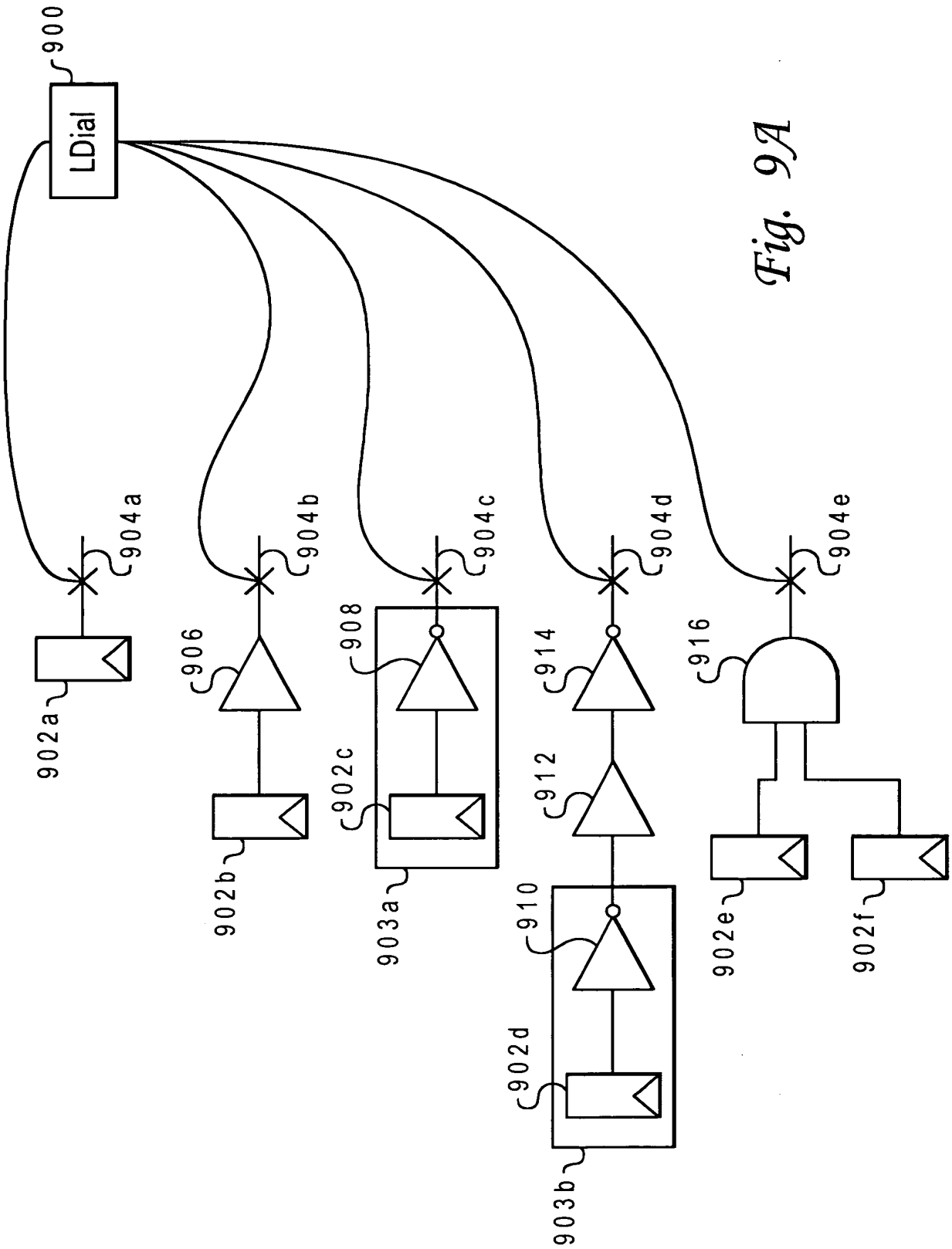
900
LDial
904a
904b
904c
904d
904e
902a
906
902b
908
902c
903a
914
912
910
902d
903b
916
902e
902f

Fig. 9A

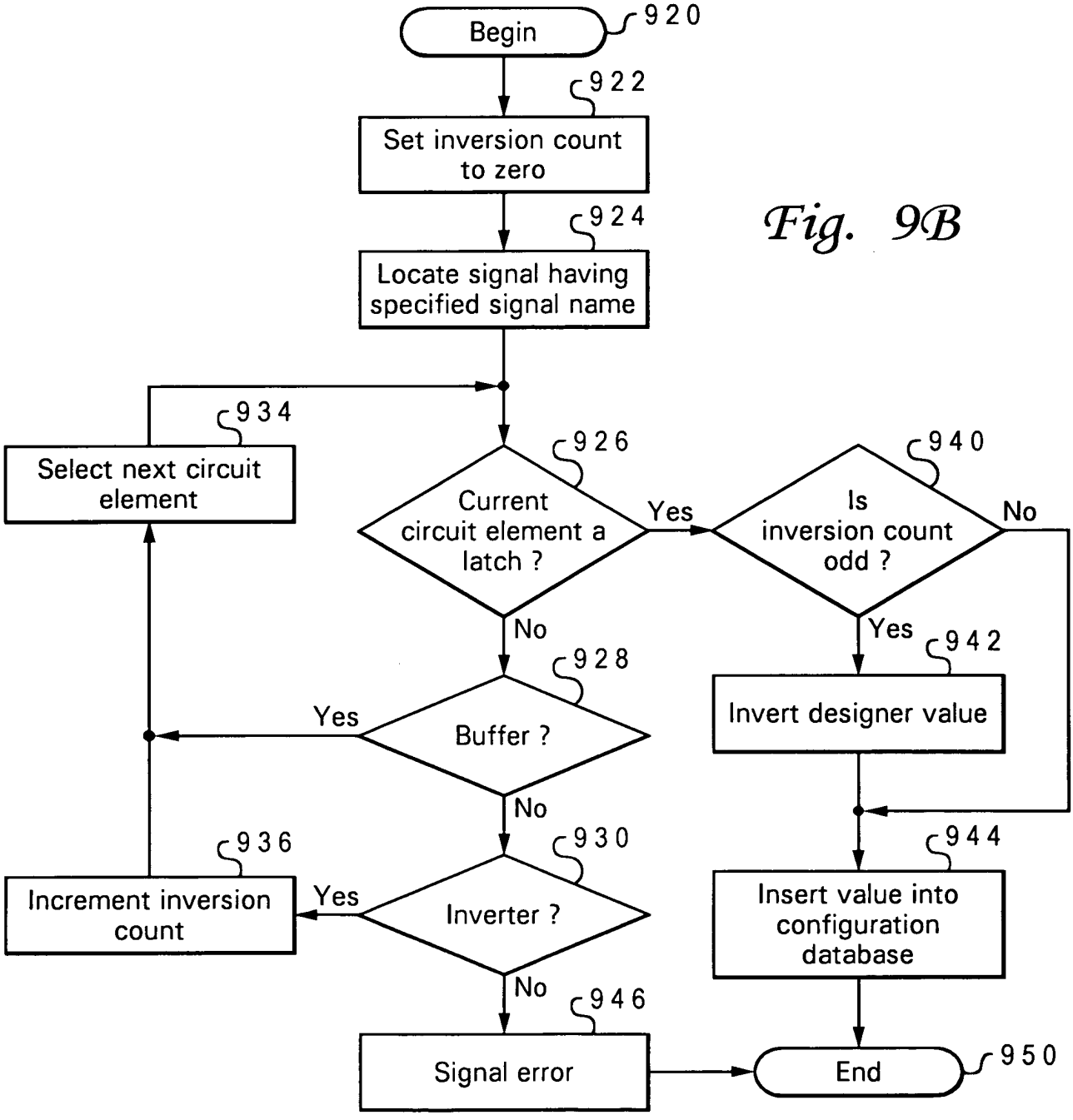
Begin
920
922
Set inversion count to zero
924
Locate signal having specified signal name
934
Select next circuit element
926
Current circuit element a latch ?
Yes
940
Is inversion count odd ?
No
No
Yes
942
Invert designer value
928
Yes
Buffer ?
No
936
Increment inversion count
Yes
930
Inverter ?
944
Insert value into configuration database
No
946
Signal error
End
950

Fig. 9B

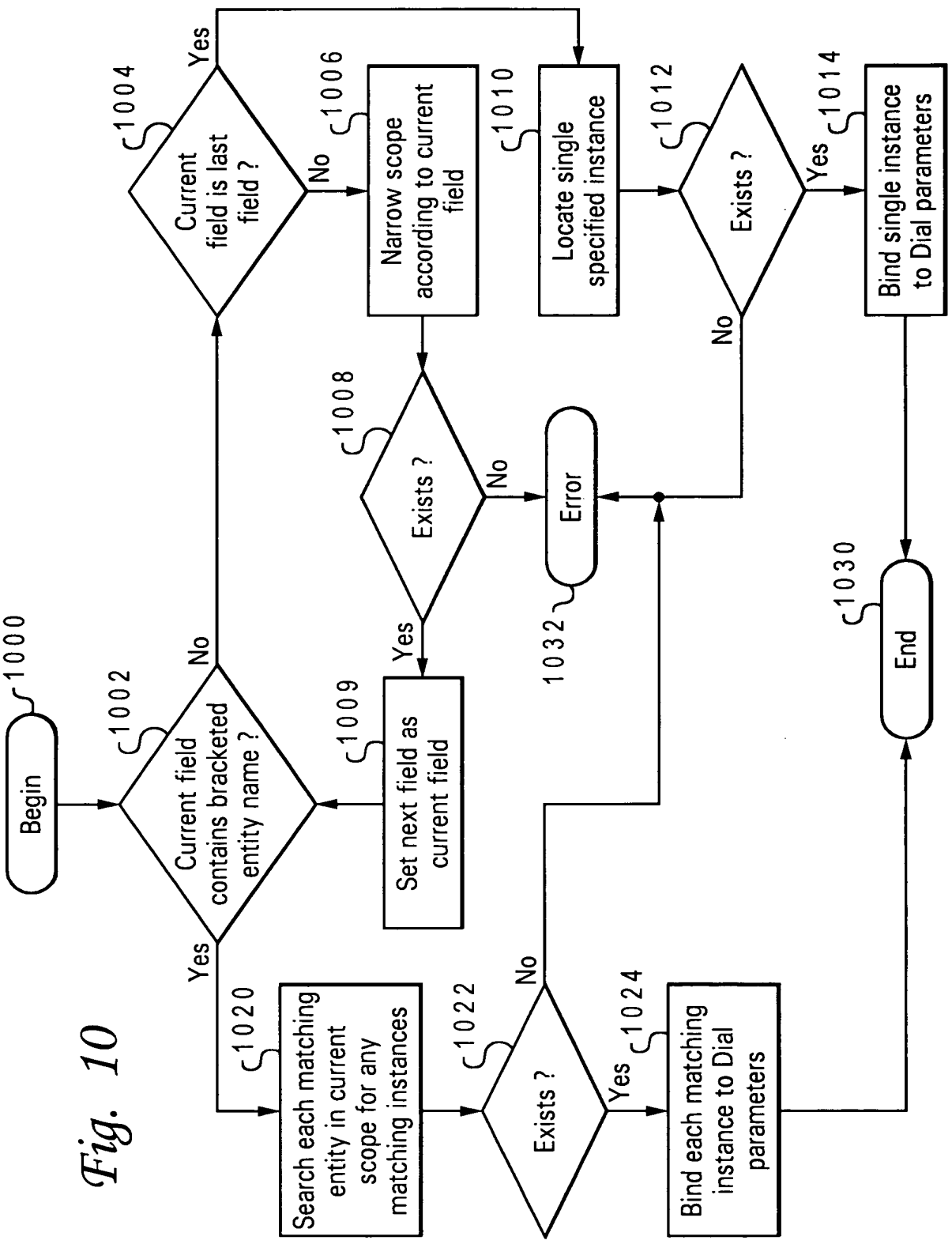
Begin
1000
Current field contains bracketed entity name ?
1002
No
Yes
Current field is last field ?
1004
Yes
No
Narrow scope according to current field
1006
Set next field as current field
1009
Exists ?
1008
Yes
No
Error
1032
Locate single specified instance
1010
Exists ?
1012
No
Yes
Bind single instance to Dial parameters
1014
End
1030
Search each matching entity in current scope for any matching instances
1020
Exists ?
1022
No
Yes
Bind each matching instance to Dial parameters
1024

Fig. 10

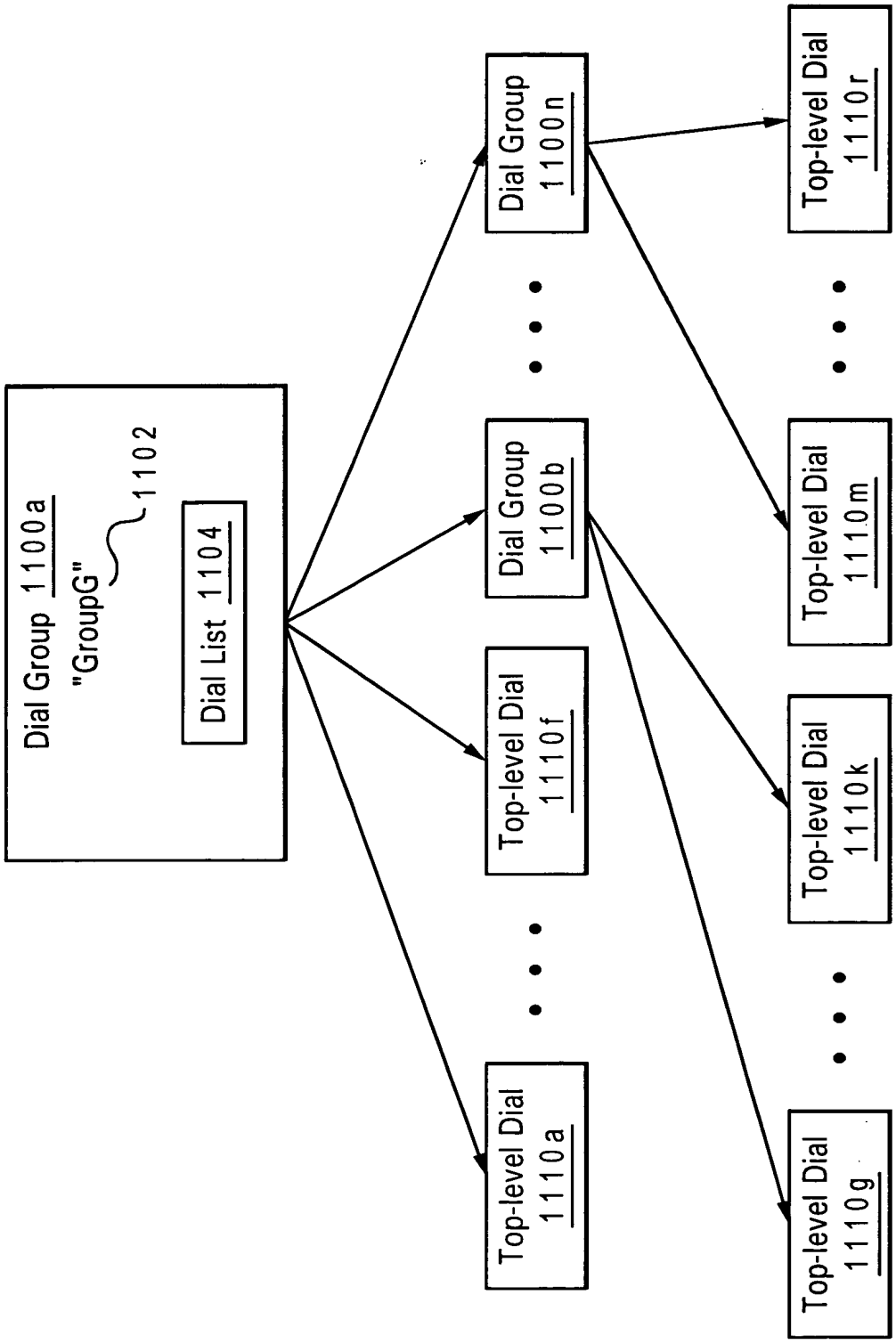
Dial Group 1100a
"GroupG"
1102
Dial List 1104
Dial Group 1100n
Dial Group 1100b
Top-level Dial 1110a
Top-level Dial 1110f
Top-level Dial 1110g
Top-level Dial 1110k
Top-level Dial 1110m
Top-level Dial 1110r

Fig. 11A

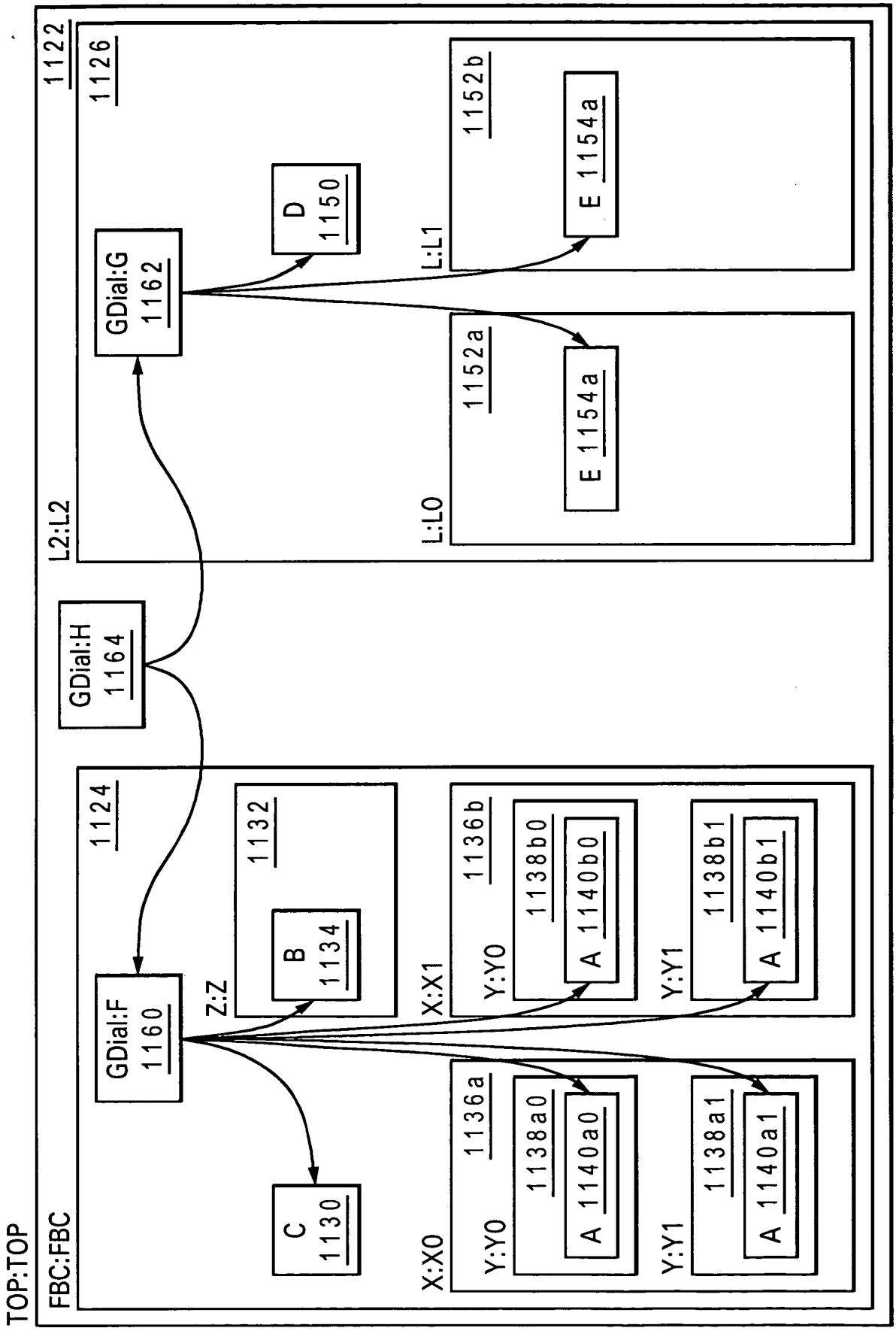
TOP:TOP
FBC:FBC
1124
1122
1126
L2:L2
GDial:F
1160
GDial:G
1162
GDial:H
1164
C
1130
1132
B
1134
Z:Z
X:X0
1136a
Y:Y0
1138a0
A 1140a0
Y:Y1
1138a1
A 1140a1
X:X1
1136b
Y:Y0
1138b0
A 1140b0
Y:Y1
1138b1
A 1140b1
D
1150
L:L0
1152a
E 1154a
L:L1
1152b
E 1154a
1120

Fig. 11B

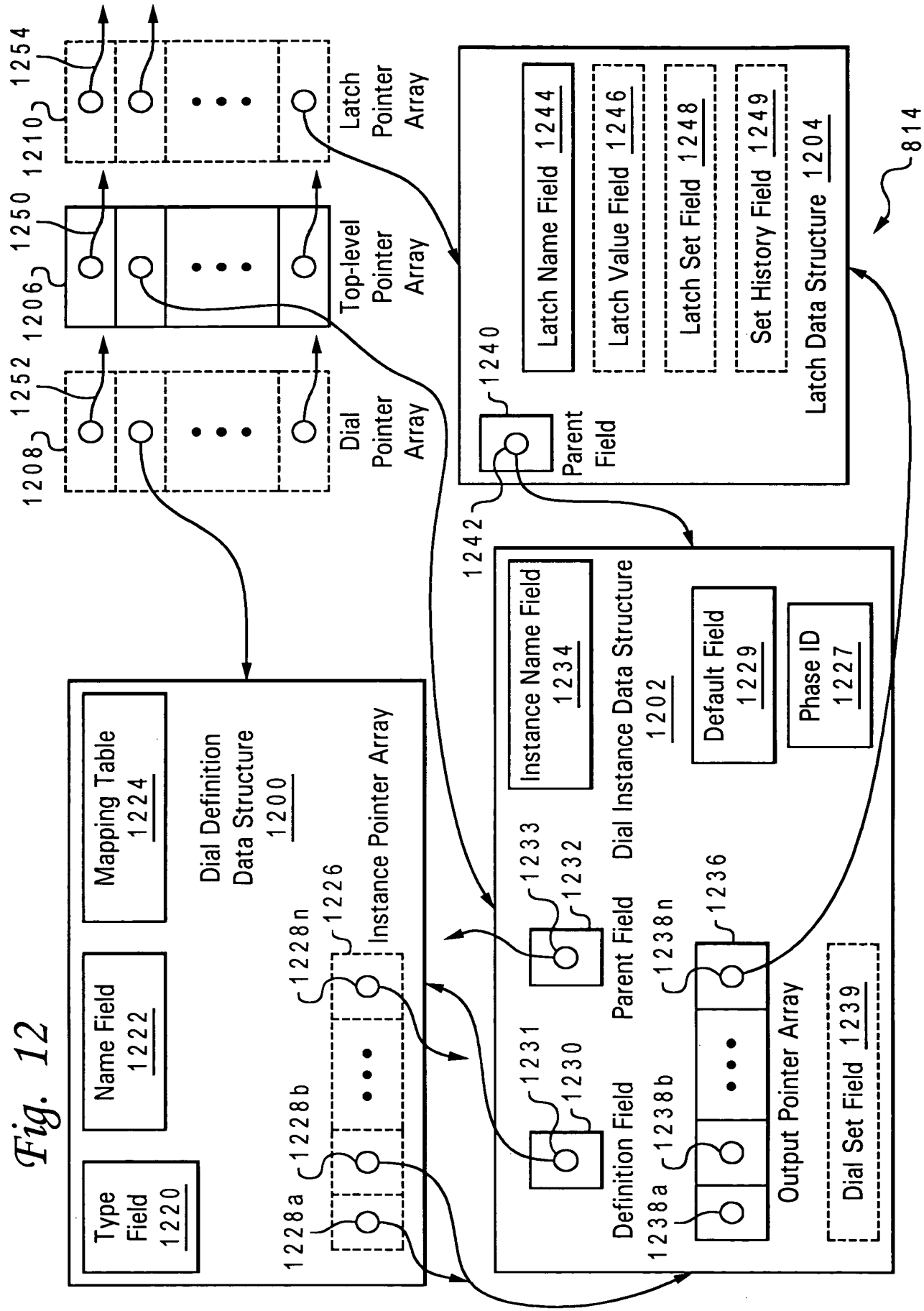
Fig. 12
Type Field 1220
Name Field 1222
Mapping Table 1224
Dial Definition Data Structure 1200
Instance Pointer Array
1226
1228a
1228b
1228n
Dial Pointer Array
1208
1252
Top-level Pointer Array
1206
1250
Latch Pointer Array
1210
1254
Latch Data Structure 1204
814
Latch Name Field 1244
Latch Value Field 1246
Latch Set Field 1248
Set History Field 1249
Parent Field
1240
1242
Dial Instance Data Structure 1202
Instance Name Field 1234
Default Field 1229
Phase ID 1227
Definition Field
1230
1231
Parent Field
1232
1233
Output Pointer Array
1236
1238a
1238b
1238n
Dial Set Field 1239

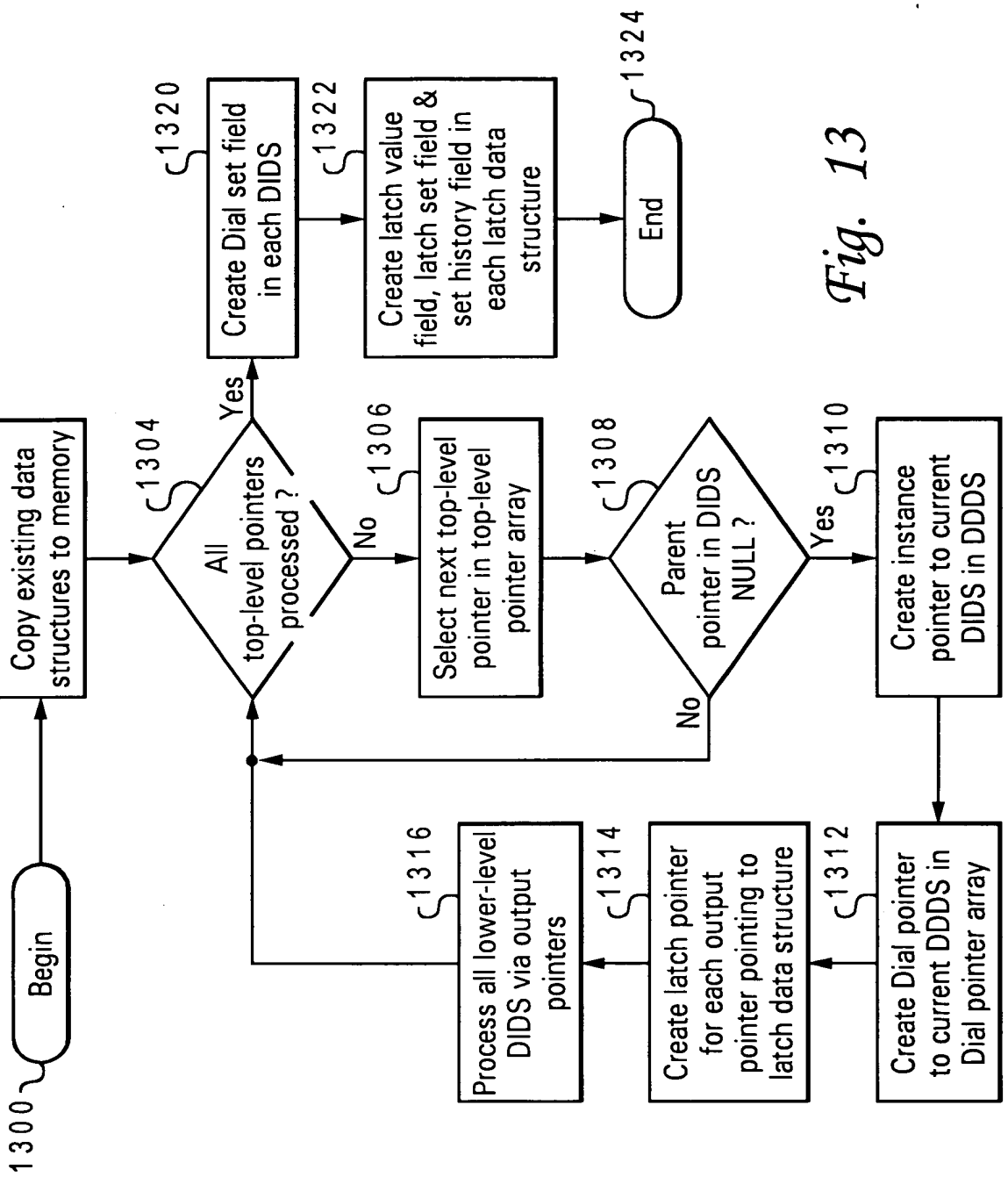
1300
Begin
1302
Copy existing data structures to memory
1304
All top-level pointers processed ?
Yes
No
1306
Select next top-level pointer in top-level pointer array
1308
Parent pointer in DIDS NULL ?
Yes
No
1310
Create instance pointer to current DIDS in DDDS
1312
Create Dial pointer to current DDDS in Dial pointer array
1314
Create latch pointer for each output pointer pointing to latch data structure
1316
Process all lower-level DIDS via output pointers
1320
Create Dial set field in each DIDS
1322
Create latch value field, latch set field & set history field in each latch data structure
1324
End

Fig. 13

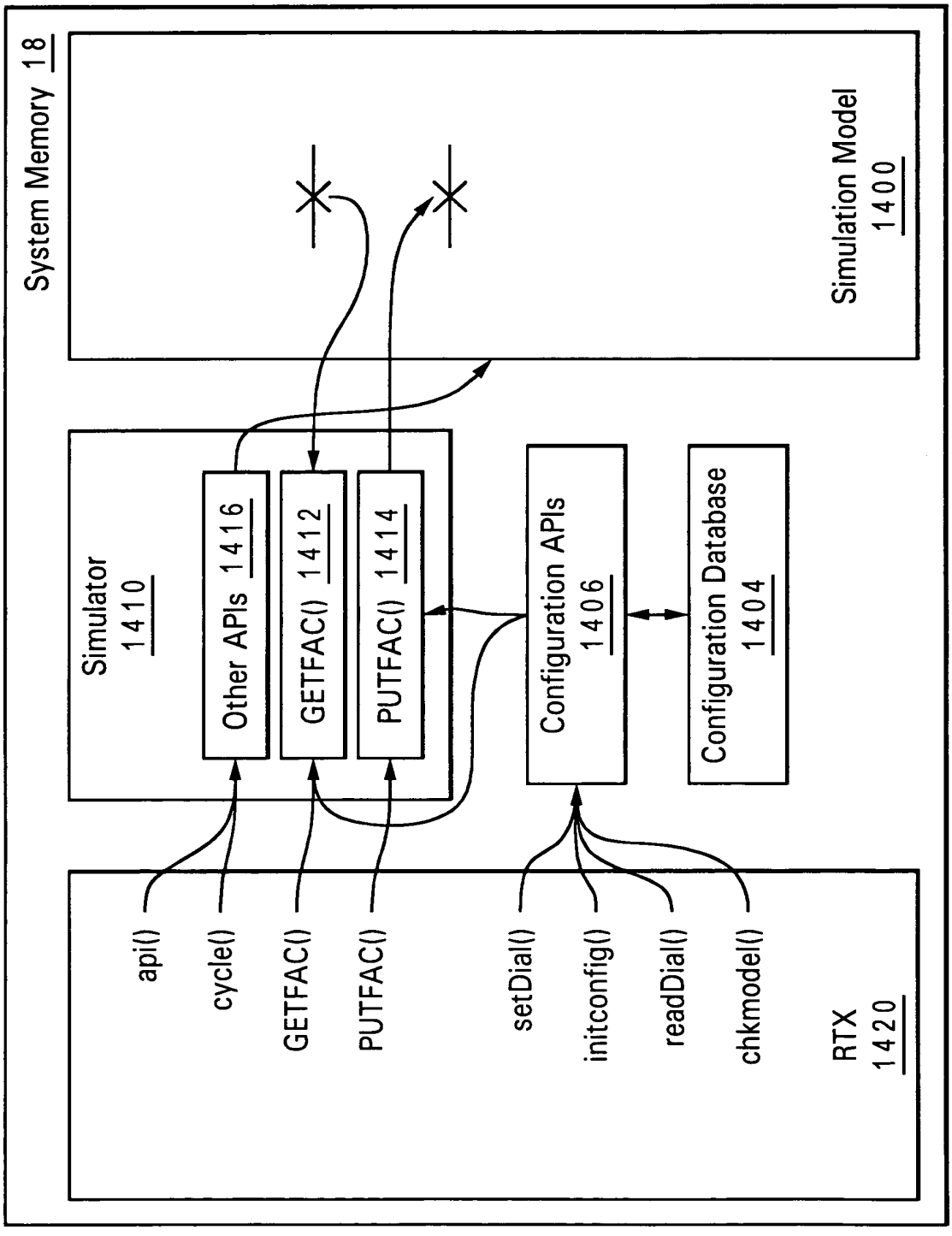
System Memory 18
Simulation Model 1400
Simulator 1410
Other APIs 1416
GETFAC() 1412
PUTFAC() 1414
Configuration APIs 1406
Configuration Database 1404
api()
cycle()
GETFAC()
PUTFAC()
setDial()
initconfig()
readDial()
chkmodel()
RTX 1420

Fig. 14

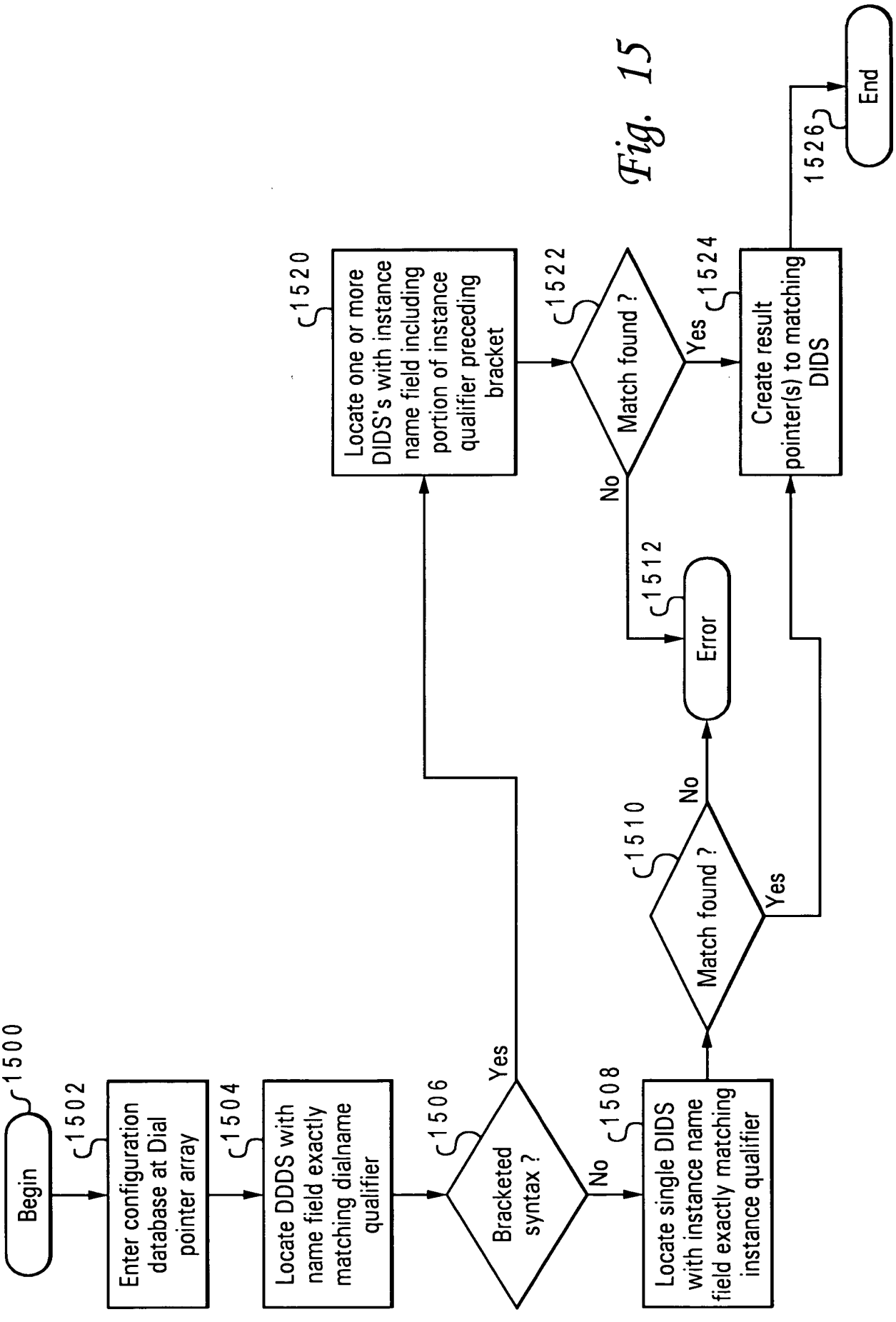
Begin
1500
Enter configuration database at Dial pointer array
1502
Locate DDDS with name field exactly matching dialname qualifier
1504
Bracketed syntax ?
1506
Yes
No
Locate single DIDS with instance name field exactly matching instance qualifier
1508
Match found ?
1510
No
Yes
Error
1512
Locate one or more DIDS's with instance name field including portion of instance qualifier preceding bracket
1520
Match found ?
1522
No
Yes
Create result pointer(s) to matching DIDS
1524
End
1526

Fig. 15

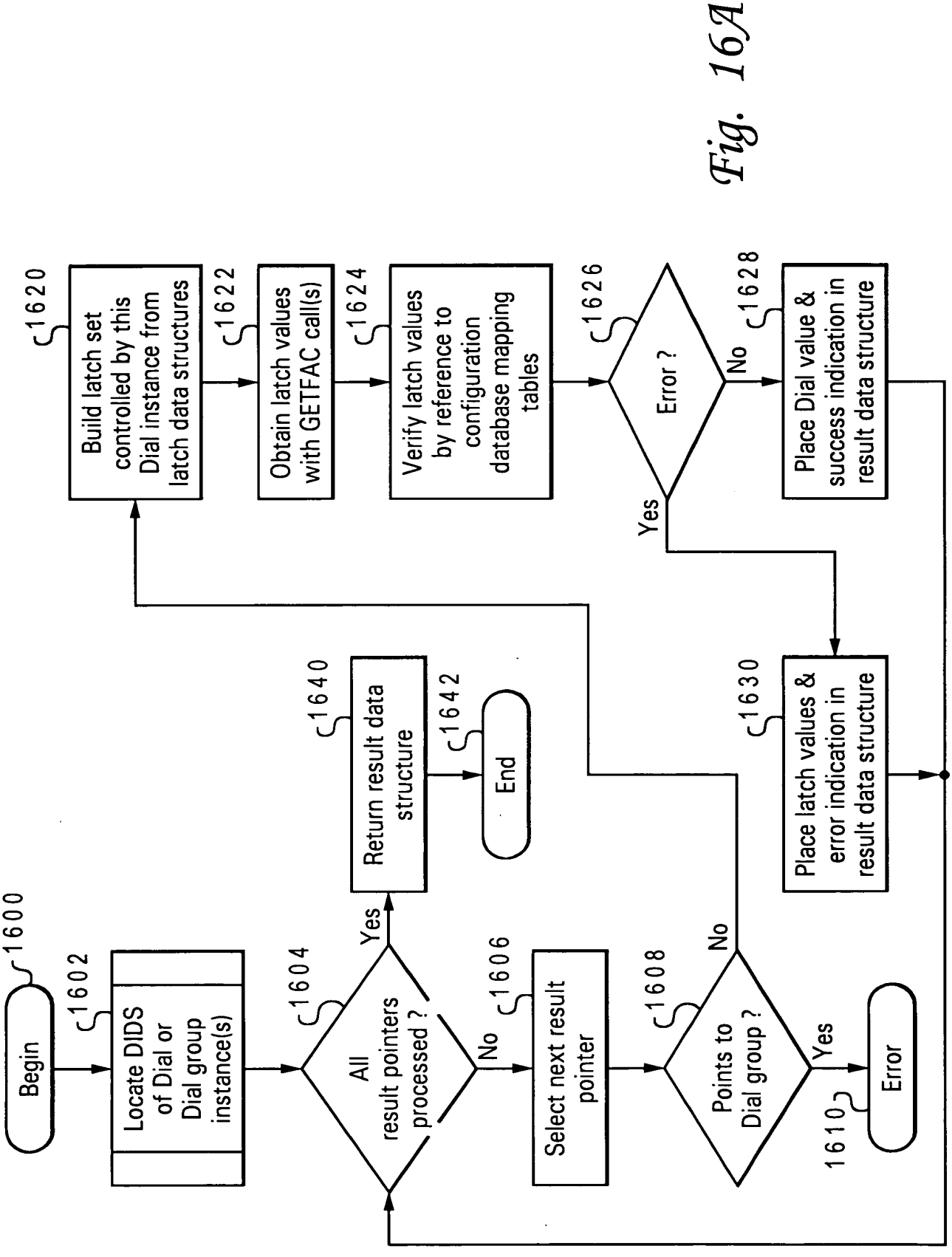
1600
Begin
1602
Locate DIDS of Dial or Dial group instance(s)
1604
All result pointers processed ?
Yes
1640
Return result data structure
1642
End
No
1606
Select next result pointer
1608
Points to Dial group ?
No
1630
Place latch values & error indication in result data structure
Yes
1610
Error
1620
Build latch set controlled by this Dial instance from latch data structures
1622
Obtain latch values with GETFAC call(s)
1624
Verify latch values by reference to configuration database mapping tables
1626
Error ?
No
1628
Place Dial value & success indication in result data structure
Yes

Fig. 16A

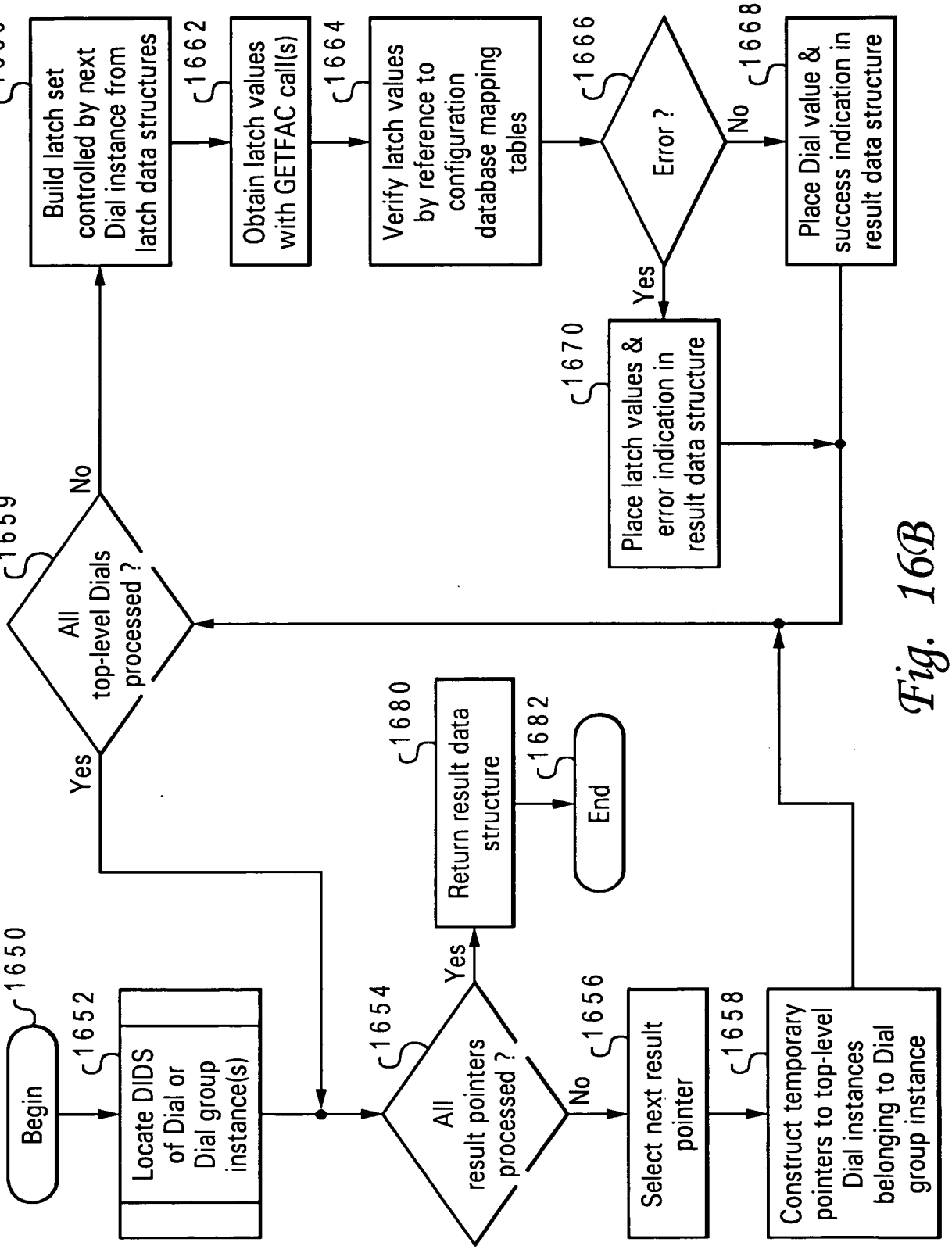
1650
Begin
1652
Locate DIDS of Dial or Dial group instance(s)
1654
All result pointers processed ?
Yes
No
1656
Select next result pointer
1658
Construct temporary pointers to top-level Dial instances belonging to Dial group instance
1659
All top-level Dials processed ?
Yes
No
1660
Build latch set controlled by next Dial instance from latch data structures
1662
Obtain latch values with GETFAC call(s)
1664
Verify latch values by reference to configuration database mapping tables
1666
Error ?
Yes
No
1668
Place Dial value & success indication in result data structure
1670
Place latch values & error indication in result data structure
1680
Return result data structure
1682
End

Fig. 16B

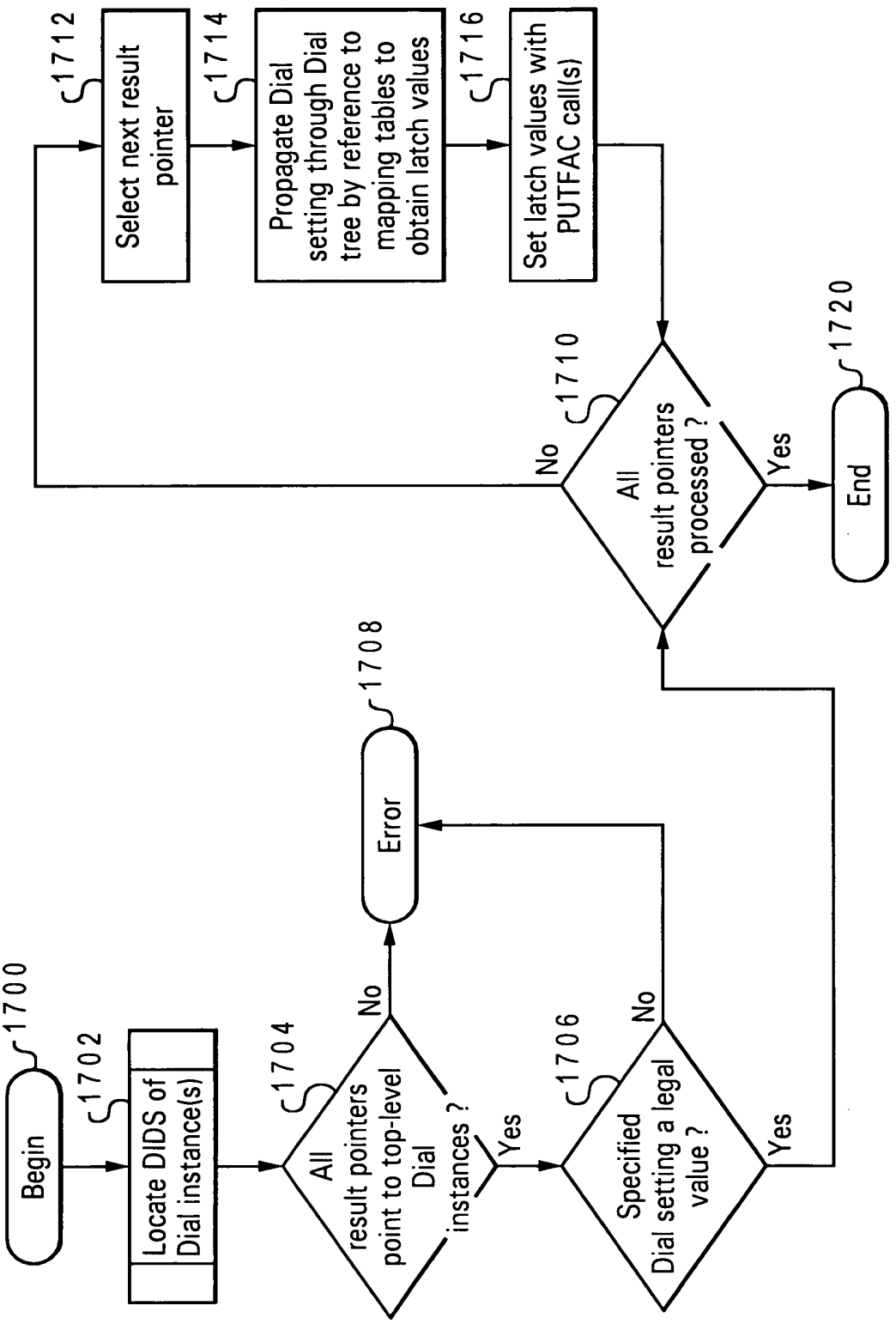
1700
Begin
1702
Locate DIDS of Dial instance(s)
1704
All result pointers point to top-level Dial instances ?
No
1708
Error
Yes
1706
Specified Dial setting a legal value ?
No
Yes
1710
All result pointers processed ?
No
1712
Select next result pointer
1714
Propagate Dial setting through Dial tree by reference to mapping tables to obtain latch values
1716
Set latch values with PUTFAC call(s)
Yes
1720
End

Fig. 17A

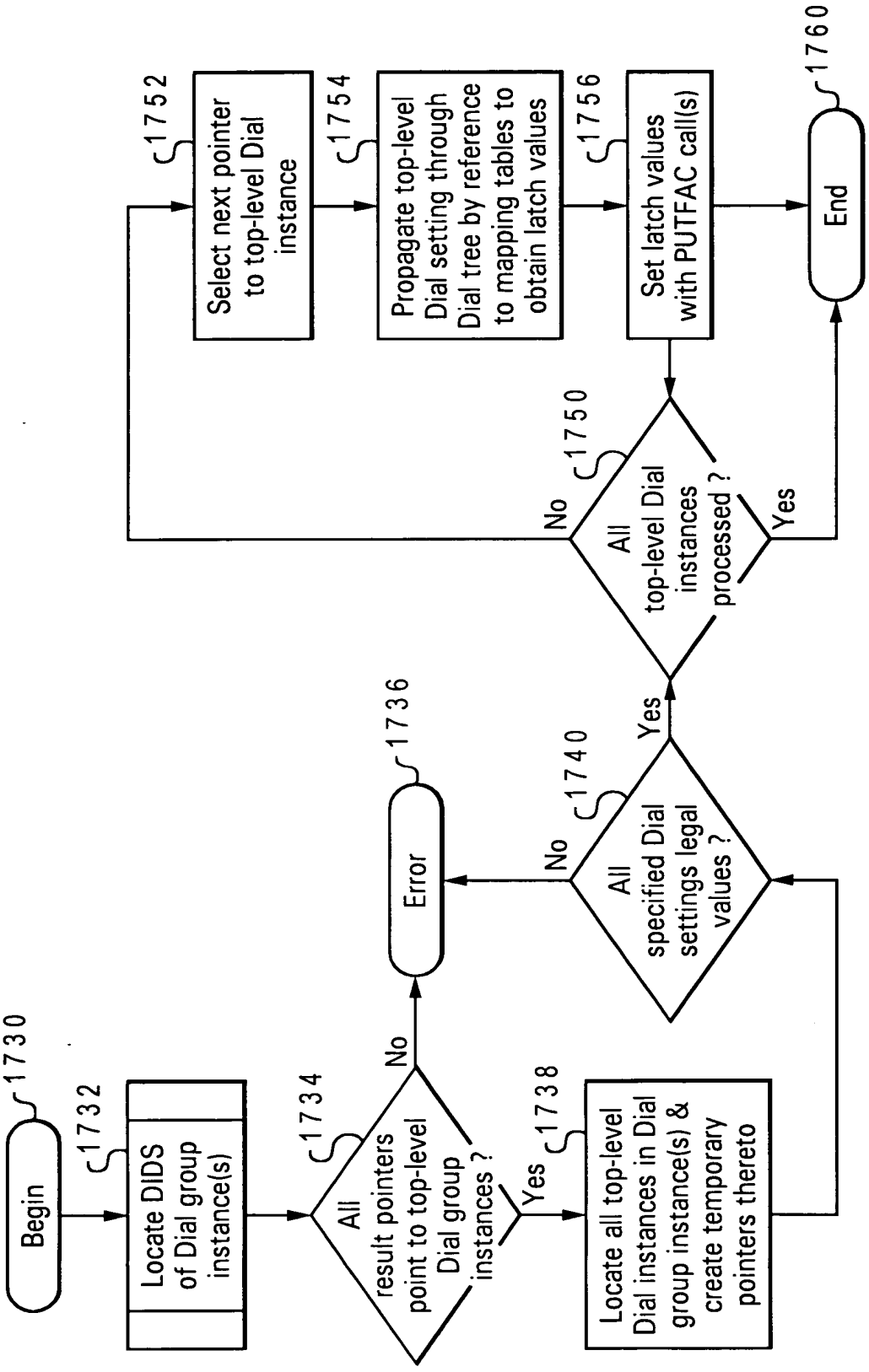
1730
Begin
1732
Locate DIDS of Dial group instance(s)
1734
All result pointers point to top-level Dial group instances ?
No
1736
Error
Yes
1738
Locate all top-level Dial instances in Dial group instance(s) & create temporary pointers thereto
1740
All specified Dial settings legal values ?
No
Yes
1750
All top-level Dial instances processed ?
No
Yes
1752
Select next pointer to top-level Dial instance
1754
Propagate top-level Dial setting through Dial tree by reference to mapping tables to obtain latch values
1756
Set latch values with PUTFAC call(s)
1760
End

Fig. 17B

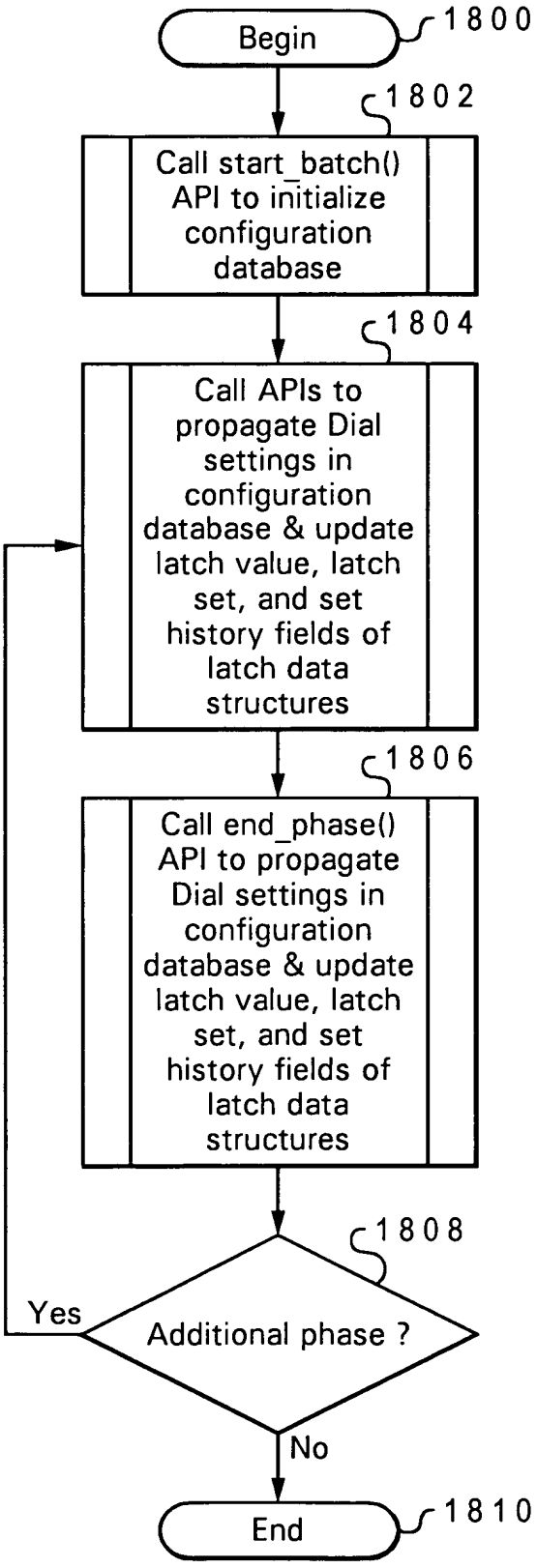
Begin
1800
1802
Call start_batch() API to initialize configuration database
1804
Call APIs to propagate Dial settings in configuration database & update latch value, latch set, and set history fields of latch data structures
1806
Call end_phase() API to propagate Dial settings in configuration database & update latch value, latch set, and set history fields of latch data structures
1808
Additional phase ?
Yes
No
End
1810

Fig. 18A

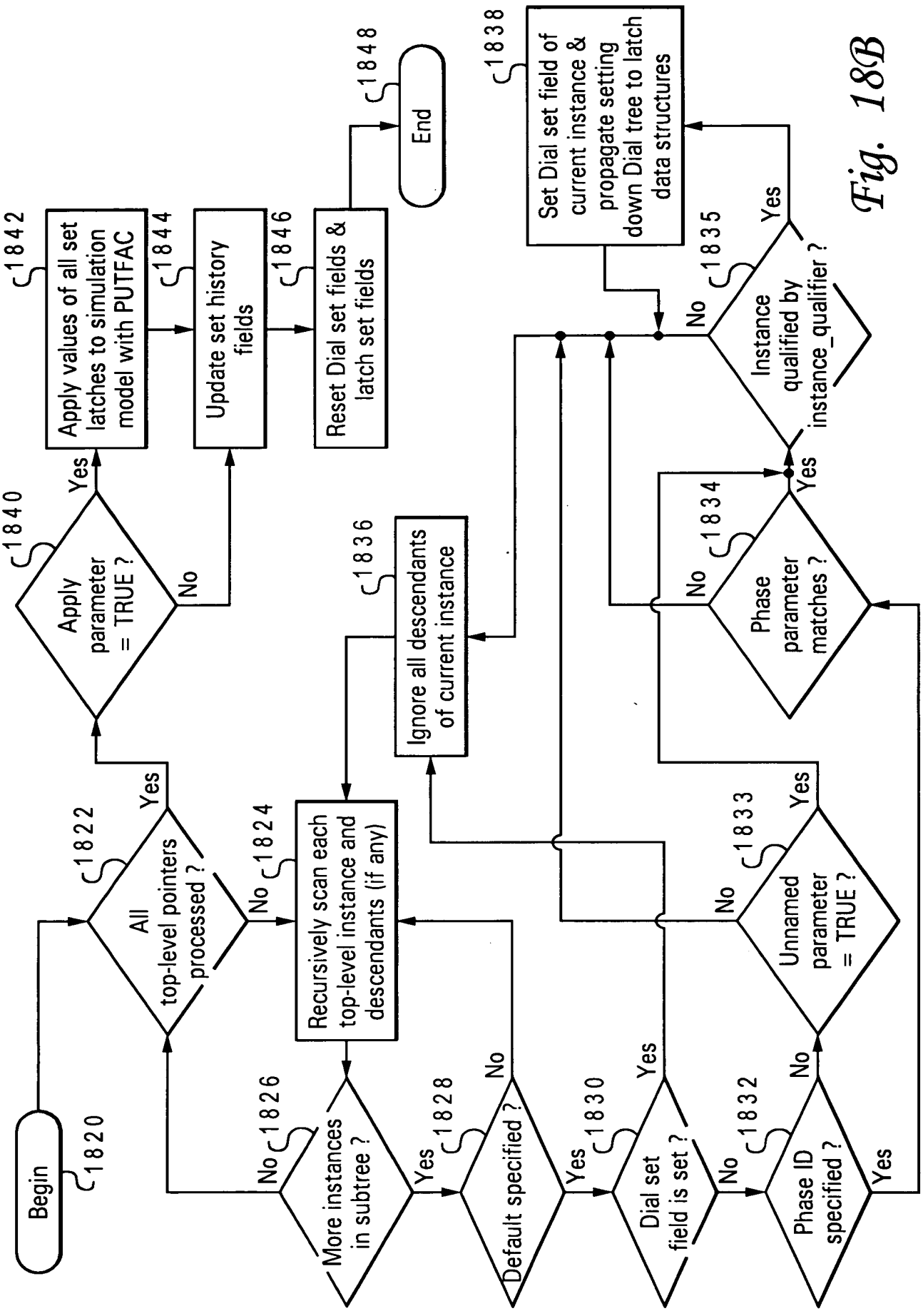
Begin
1820
All top-level pointers processed ?
1822
Yes
No
1824
Recursively scan each top-level instance and descendants (if any)
More instances in subtree ?
1826
No
Yes
Default specified ?
1828
No
Yes
Dial set field is set ?
1830
Yes
No
Phase ID specified ?
1832
No
Yes
Unnamed parameter = TRUE ?
1833
Yes
No
Phase parameter matches ?
1834
No
Yes
Instance qualified by instance_qualifier ?
1835
Yes
No
Set Dial set field of current instance & propagate setting down Dial tree to latch data structures
1838
Ignore all descendants of current instance
1836
Apply parameter = TRUE ?
1840
Yes
No
Apply values of all set latches to simulation model with PUTFAC
1842
Update set history fields
1844
Reset Dial set fields & latch set fields
1846
End
1848

Fig. 18B

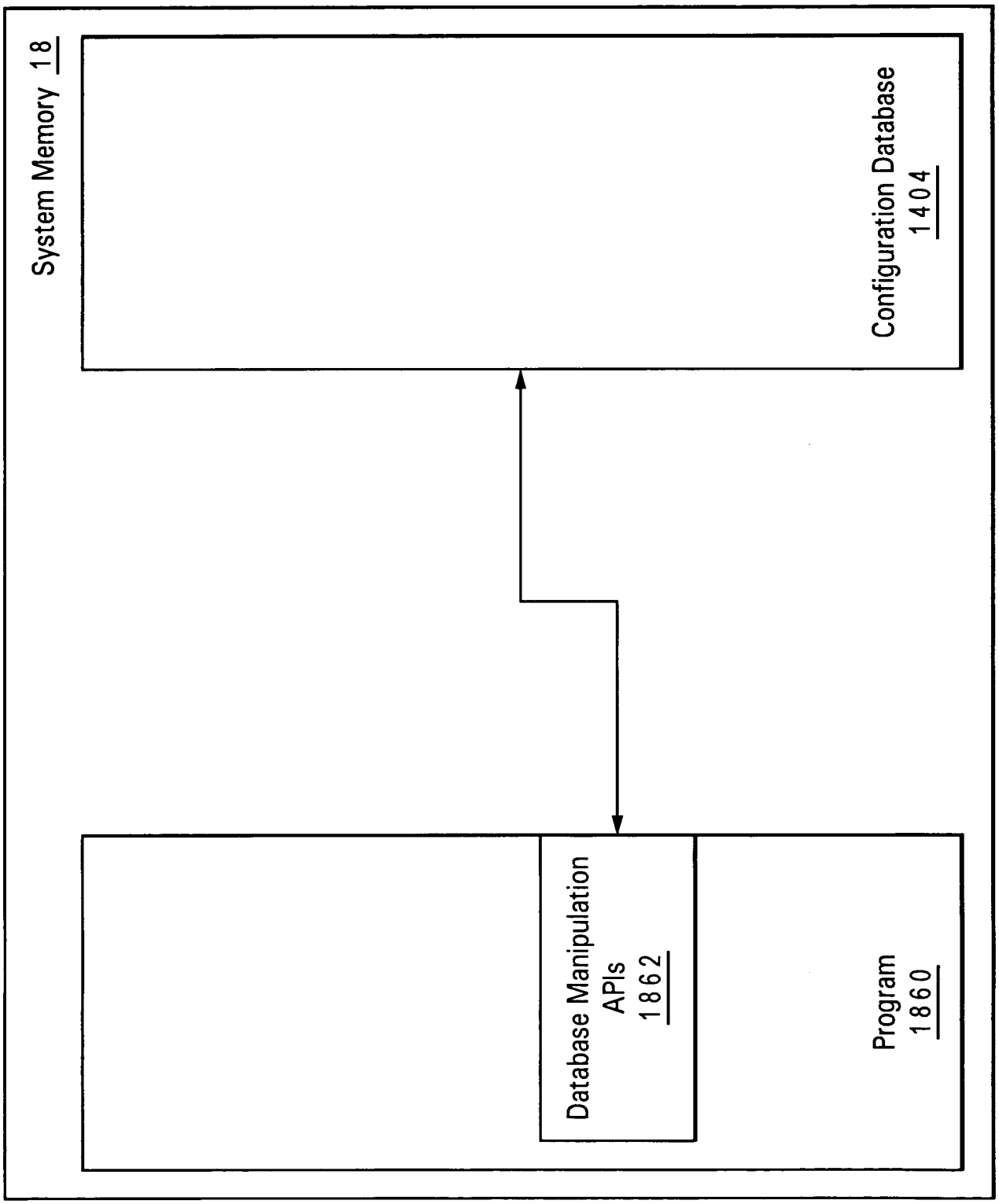
System Memory 18
Configuration Database 1404
Database Manipulation APIs 1862
Program 1860

Fig. 18C

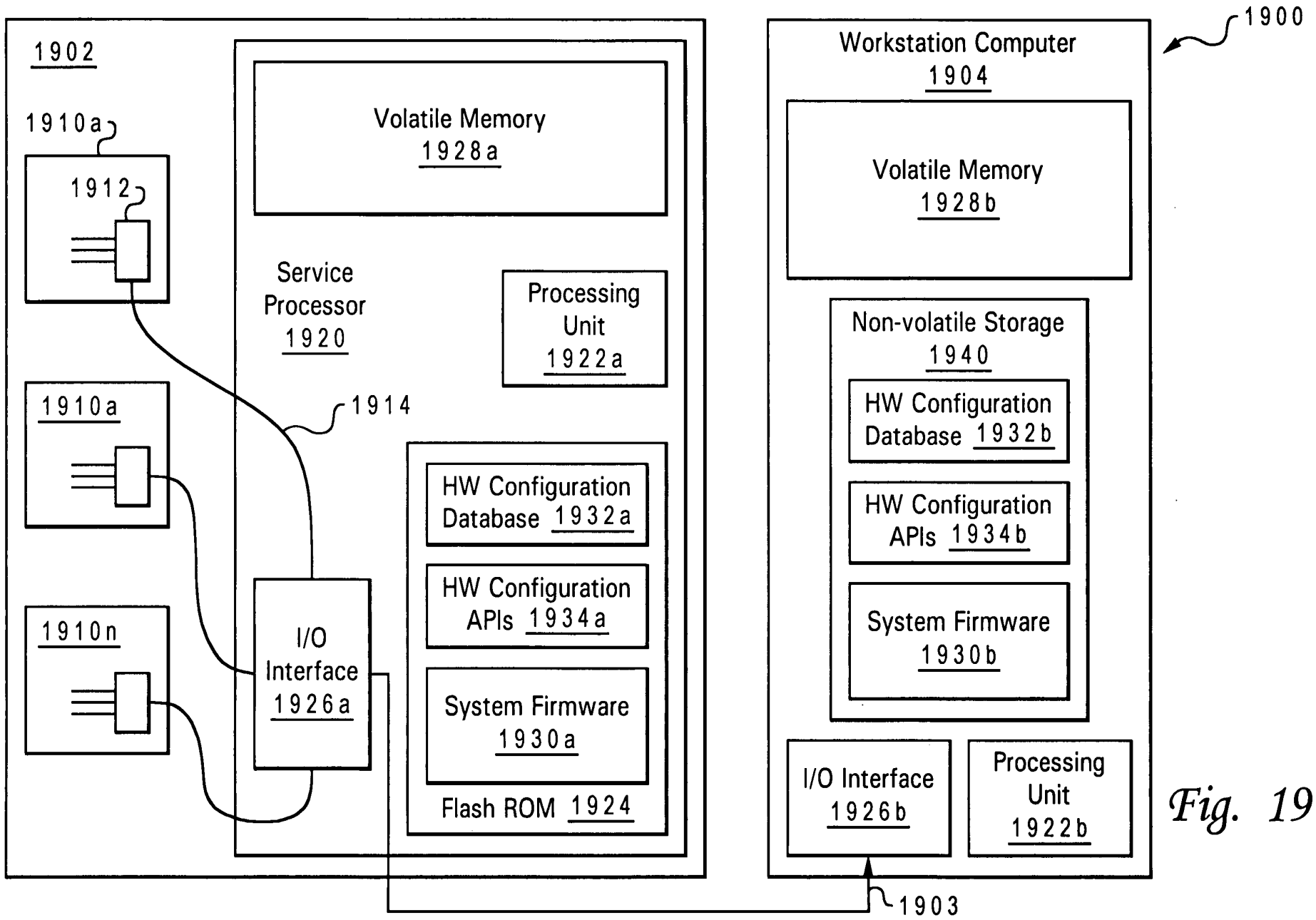
1902
1910a
1912
1910a
1910n
Volatile Memory
1928a
Service
Processor
1920
Processing
Unit
1922a
1914
HW Configuration
Database 1932a
HW Configuration
APIs 1934a
System Firmware
1930a
Flash ROM 1924
I/O
Interface
1926a
1900
Workstation Computer
1904
Volatile Memory
1928b
Non-volatile Storage
1940
HW Configuration
Database 1932b
HW Configuration
APIs 1934b
System Firmware
1930b
I/O Interface
1926b
Processing
Unit
1922b
1903

Fig. 19

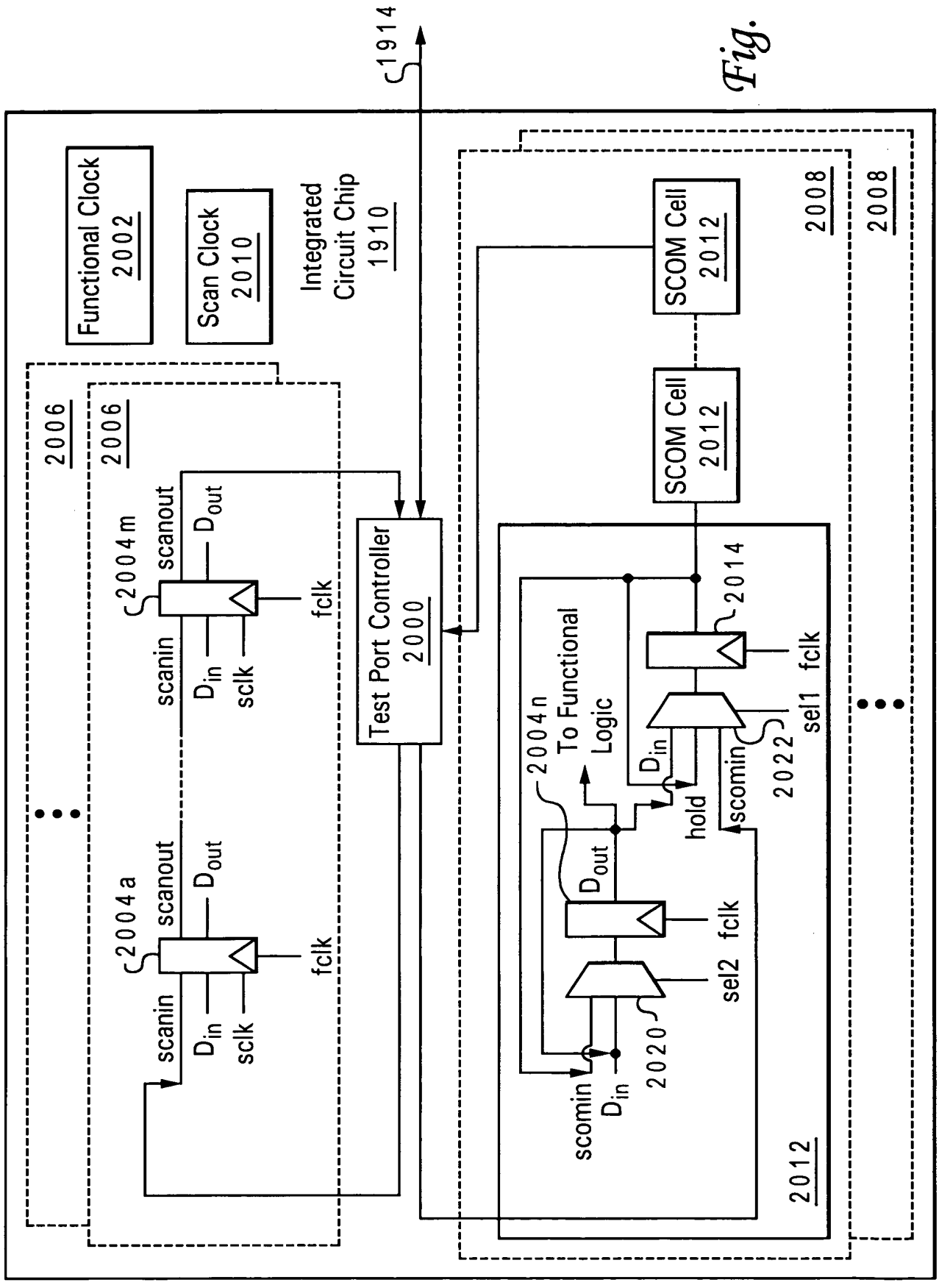
1914
Functional Clock
2002
Scan Clock
2010
Integrated
Circuit Chip
1910
2006
2006
2004m
scanout
D_{out}
scanin
D_{in}
sclk
fclk
2004a
Test Port Controller
2000
SCOM Cell
2012
SCOM Cell
2012
2008
2008
2014
2004n
To Functional
Logic
D_{in}
hold
scomin
2022
sel1
fclk
D_{out}
sel2
fclk
2020
scomin
D_{in}
2012

Fig. 20

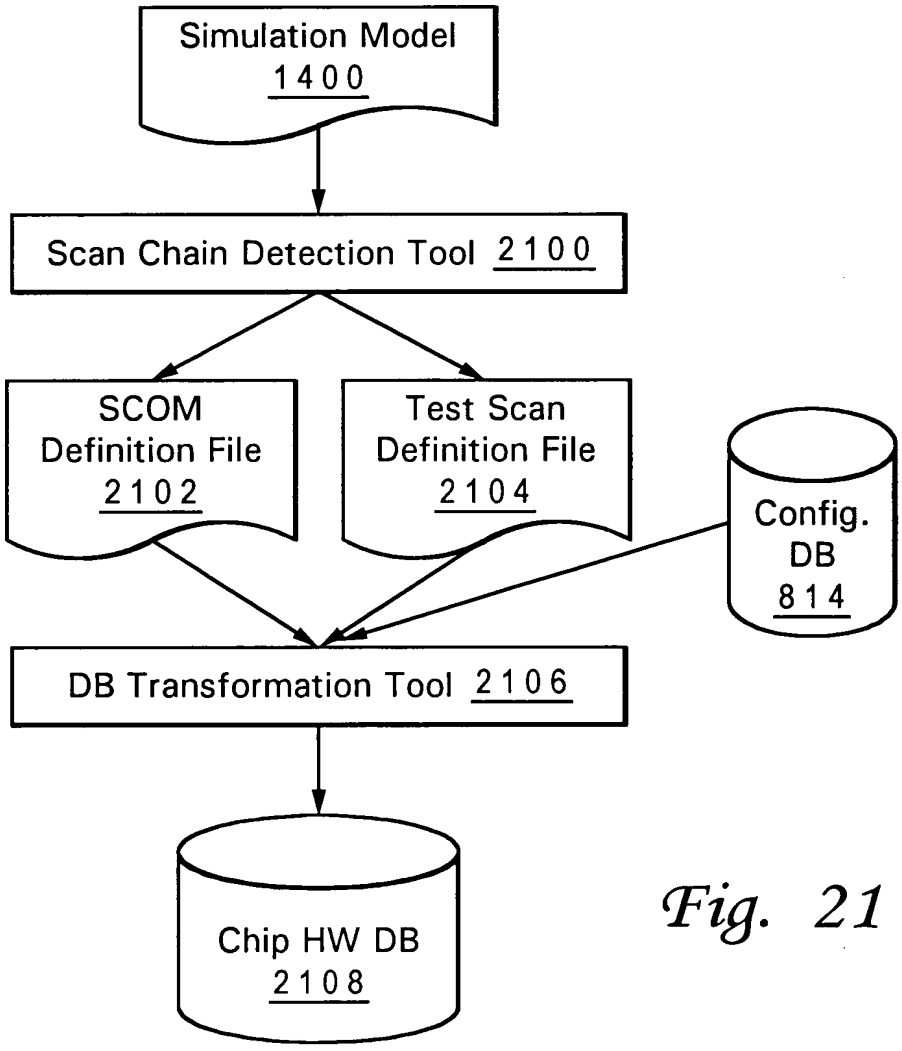
Simulation Model
1400
Scan Chain Detection Tool 2100
SCOM
Definition File
2102
Test Scan
Definition File
2104
Config.
DB
814
DB Transformation Tool 2106
Chip HW DB
2108

Fig. 21

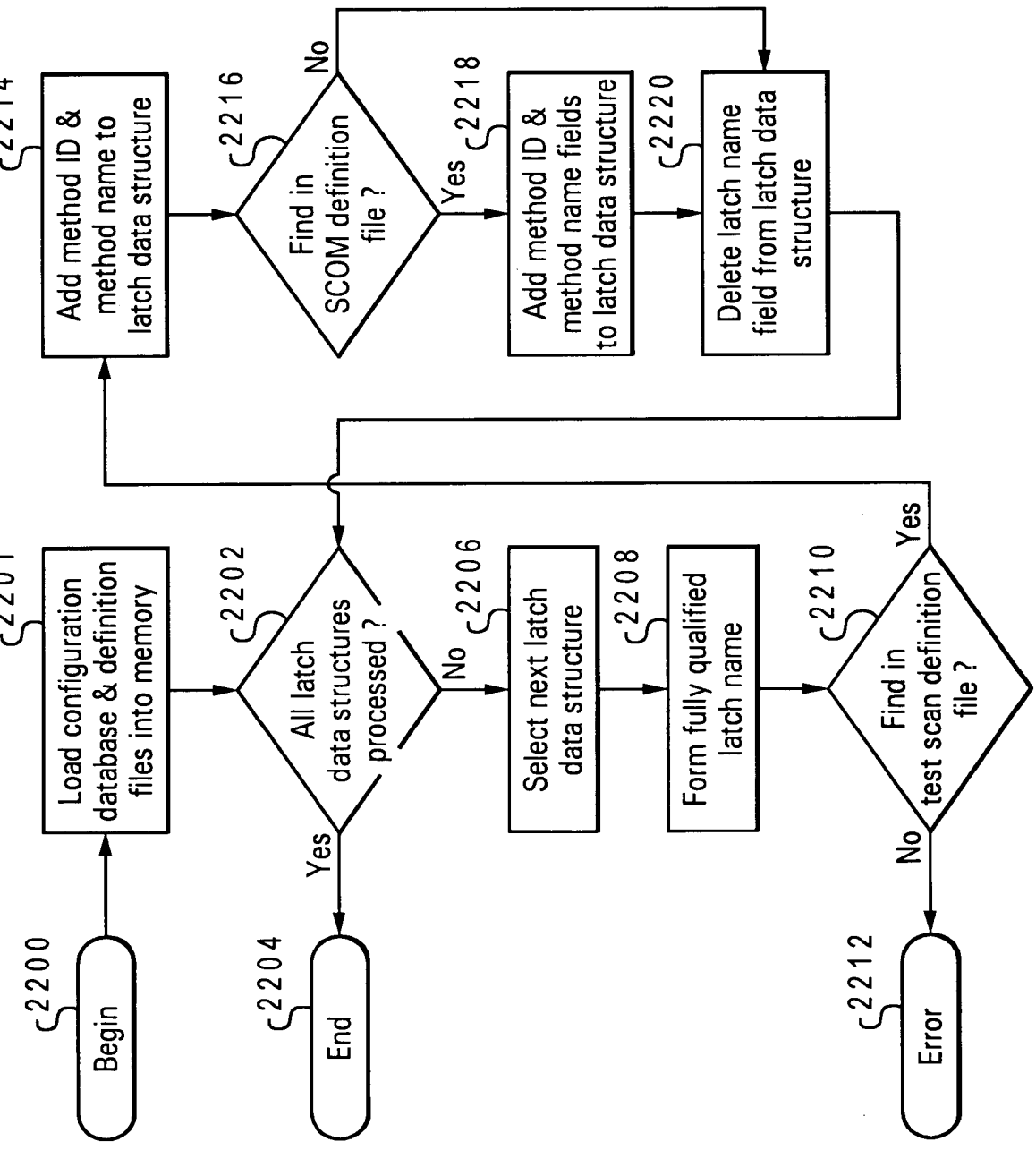
2200
Begin
2201
Load configuration database & definition files into memory
2202
All latch data structures processed ?
Yes
2204
End
No
2206
Select next latch data structure
2208
Form fully qualified latch name
2210
Find in test scan definition file ?
Yes
No
2212
Error
2214
Add method ID & method name to latch data structure
2216
Find in SCOM definition file ?
No
Yes
2218
Add method ID & method name fields to latch data structure
2220
Delete latch name field from latch data structure

Fig. 22A

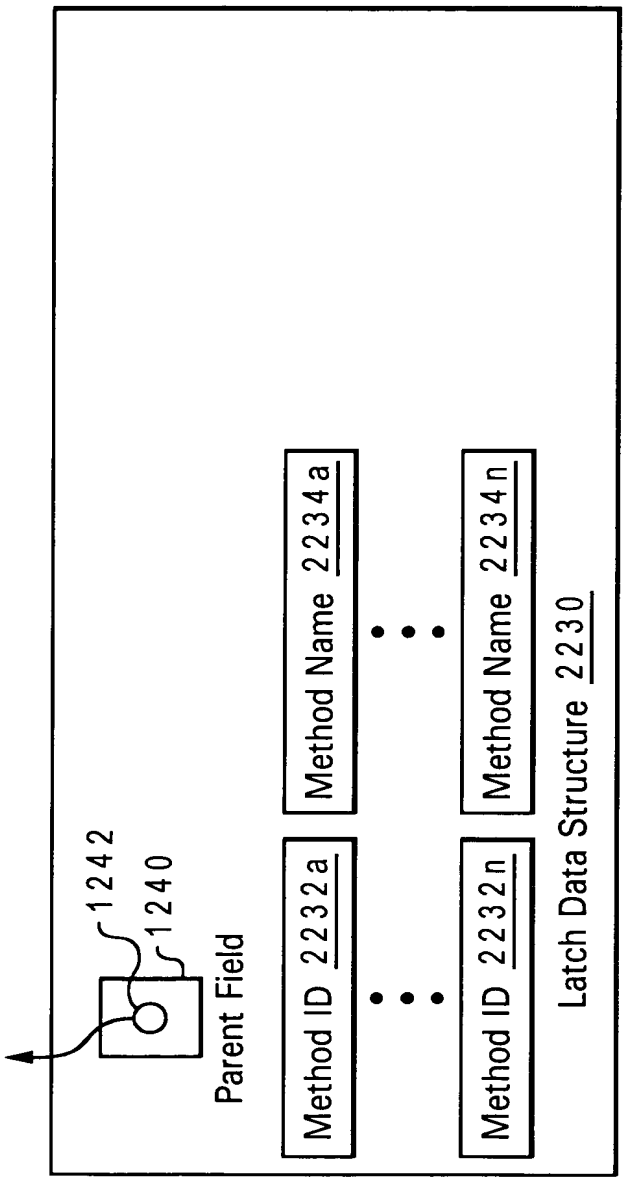
1242
1240
Parent Field
Method ID 2232a
Method ID 2232n
Method Name 2234a
Method Name 2234n
Latch Data Structure 2230

Fig. 22B

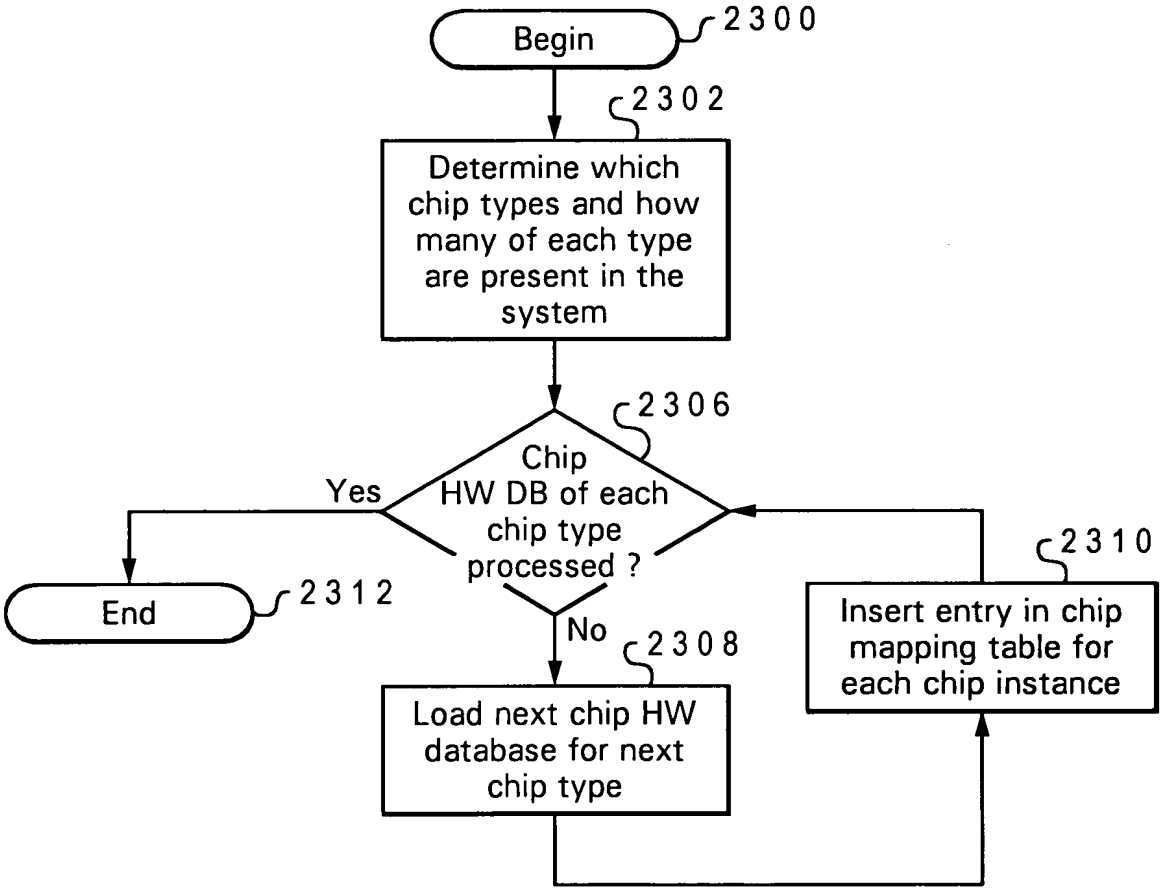
Begin
2300
2302
Determine which chip types and how many of each type are present in the system
2306
Chip HW DB of each chip type processed ?
Yes
No
End
2312
2308
Load next chip HW database for next chip type
2310
Insert entry in chip mapping table for each chip instance

Fig. 23A

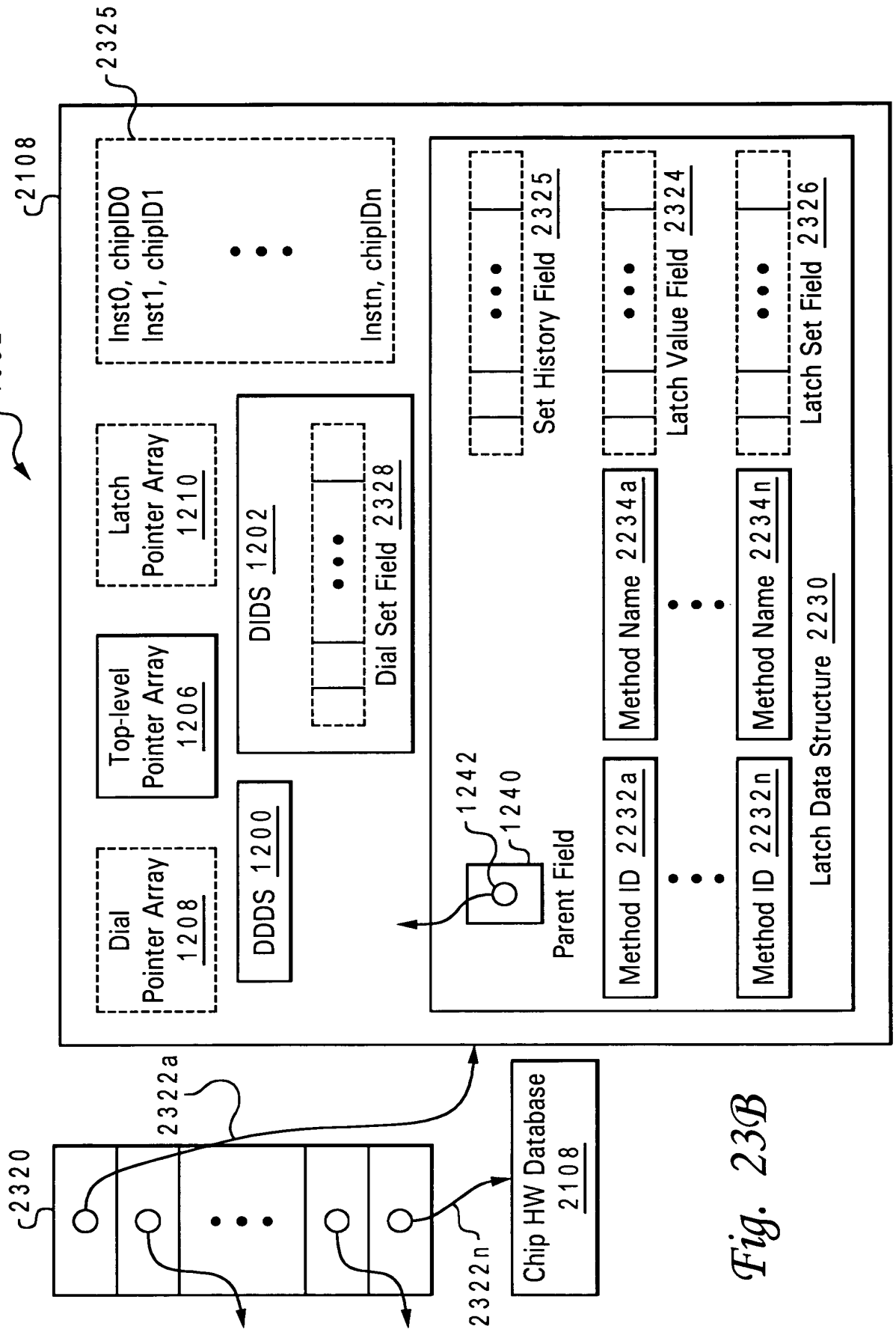
1932
2108
2325
Inst0, chipID0
Inst1, chipID1
Instn, chipIDn
Latch Pointer Array 1210
Top-level Pointer Array 1206
Dial Pointer Array 1208
DDDS 1200
DIDS 1202
Dial Set Field 2328
1242
1240
Parent Field
Set History Field 2325
Latch Value Field 2324
Latch Set Field 2326
Method Name 2234a
Method Name 2234n
Method ID 2232a
Method ID 2232n
Latch Data Structure 2230
2320
2322a
2322n
Chip HW Database 2108

Fig. 23B

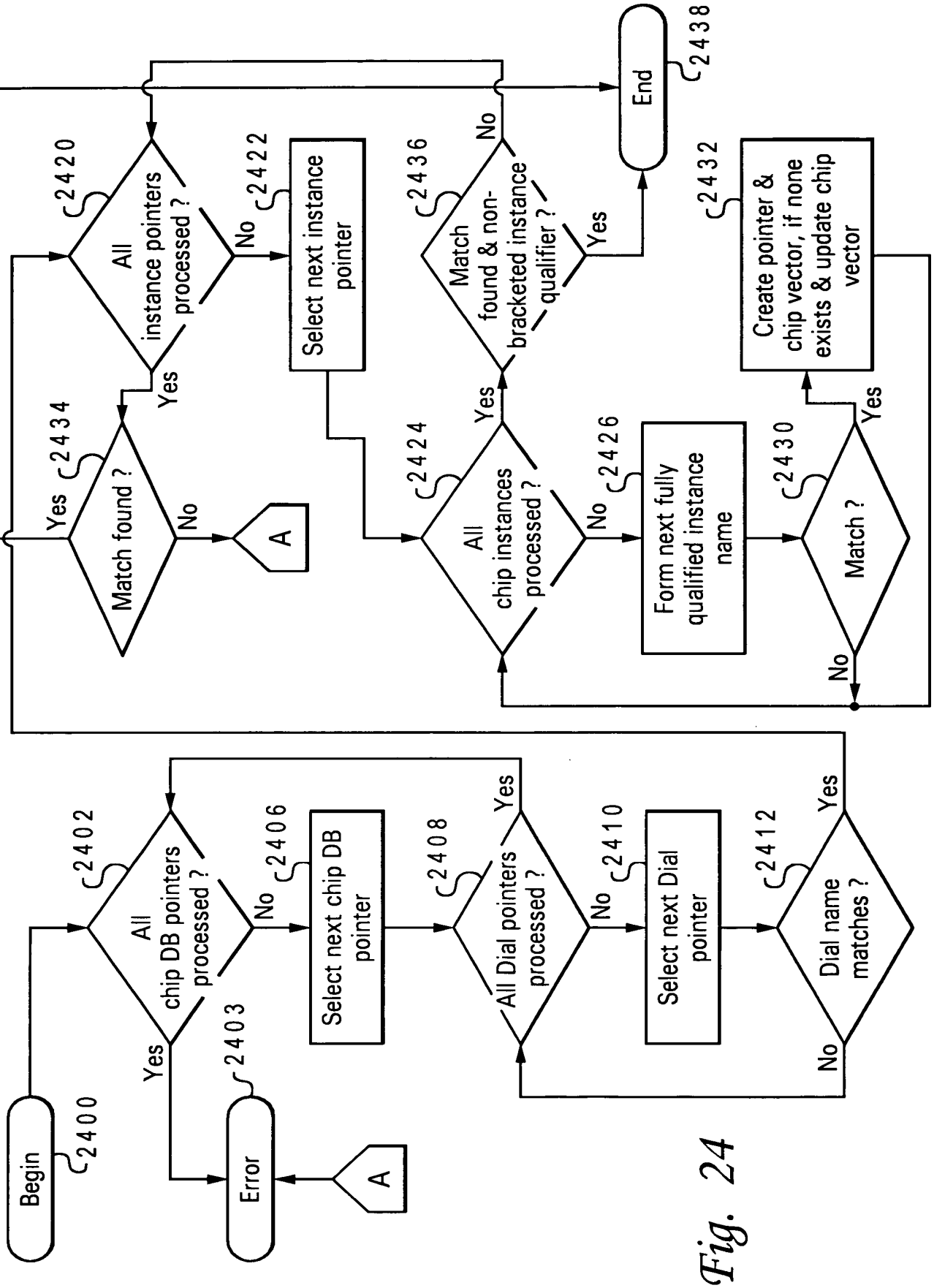
Begin
2400
All chip DB pointers processed ?
2402
Yes
Error
2403
No
Select next chip DB pointer
2406
All Dial pointers processed ?
2408
Yes
No
Select next Dial pointer
2410
Dial name matches ?
2412
Yes
No
A
All instance pointers processed ?
2420
Yes
No
Select next instance pointer
2422
All chip instances processed ?
2424
Yes
No
Form next fully qualified instance name
2426
Match found & non-bracketed instance qualifier ?
2436
Yes
No
End
2438
Match found ?
2434
Yes
No
A
Match ?
2430
Yes
No
Create pointer & chip vector, if none exists & update chip vector
2432

Fig. 24

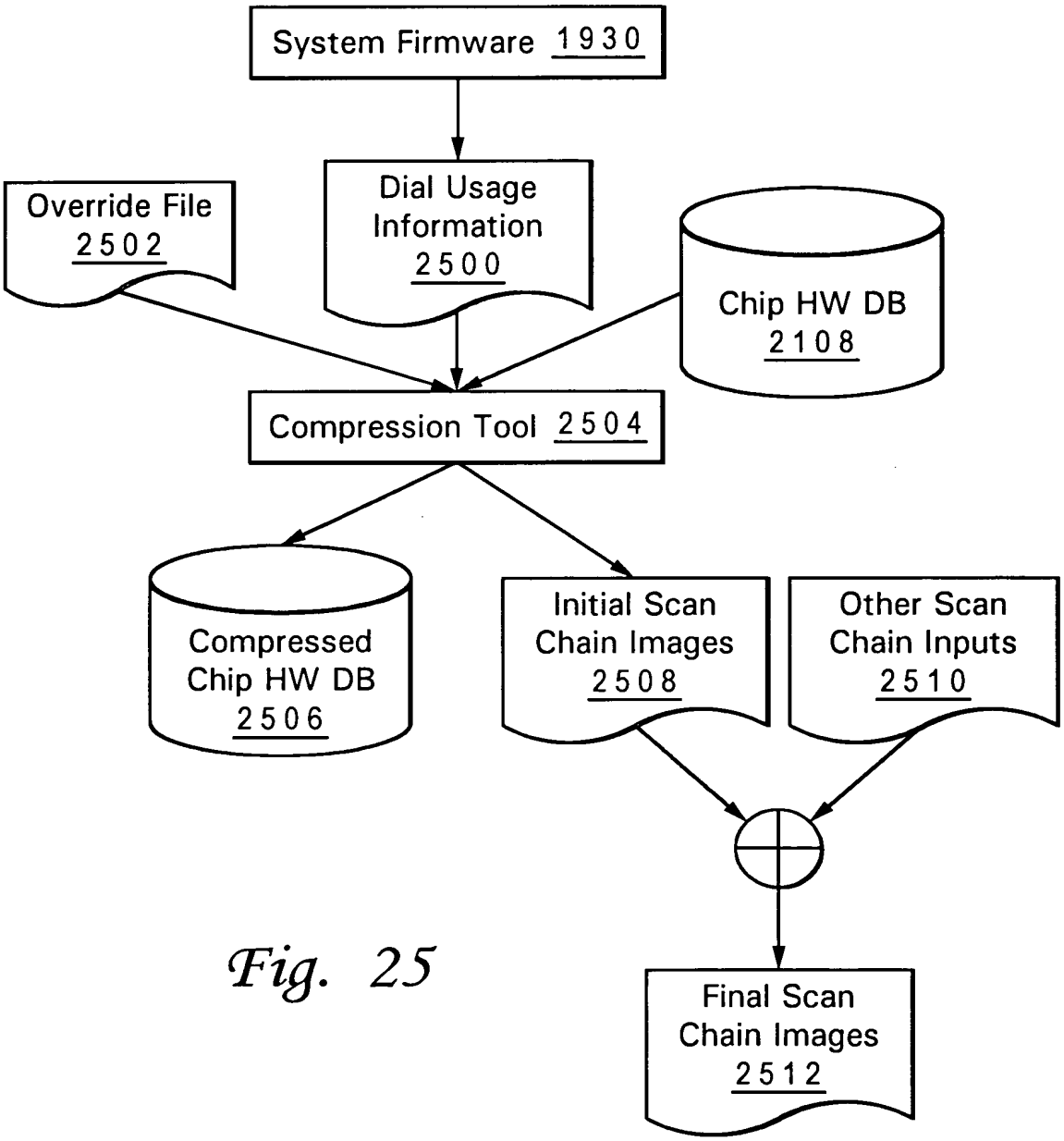
System Firmware 1930
Override File 2502
Dial Usage Information 2500
Chip HW DB 2108
Compression Tool 2504
Compressed Chip HW DB 2506
Initial Scan Chain Images 2508
Other Scan Chain Inputs 2510
Final Scan Chain Images 2512

Fig. 25

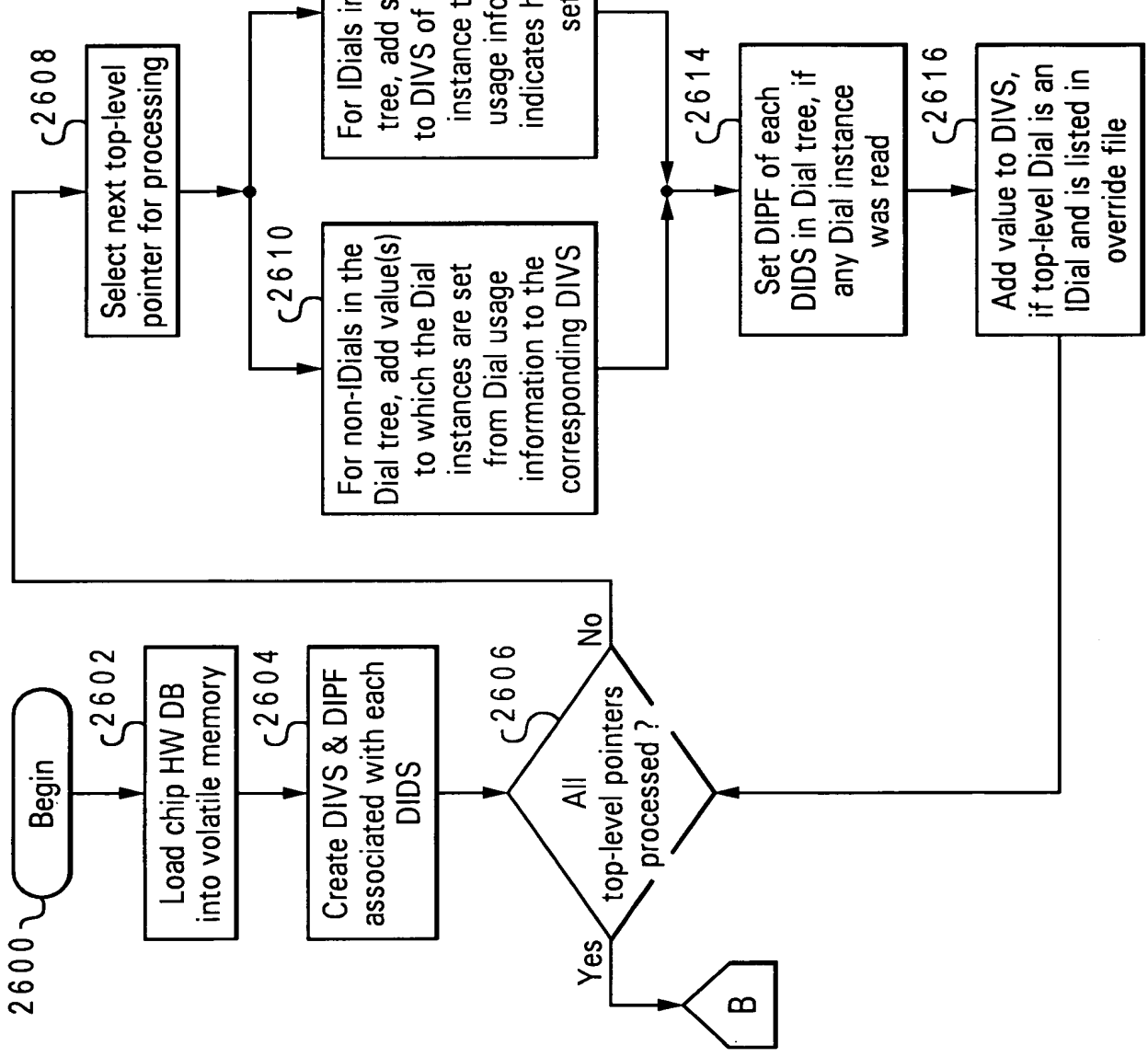
2600
Begin
2602
Load chip HW DB into volatile memory
2604
Create DIVS & DIPF associated with each DIDS
2606
All top-level pointers processed ?
No
Yes
B
2608
Select next top-level pointer for processing
2610
For non-IDials in the Dial tree, add value(s) to which the Dial instances are set from Dial usage information to the corresponding DIVS
2612
For IDials in the Dial tree, add set token to DIVS of each Dial instance that Dial usage information indicates has been set
2614
Set DIPF of each DIDS in Dial tree, if any Dial instance was read
2616
Add value to DIVS, if top-level Dial is an IDial and is listed in override file

Fig. 26A

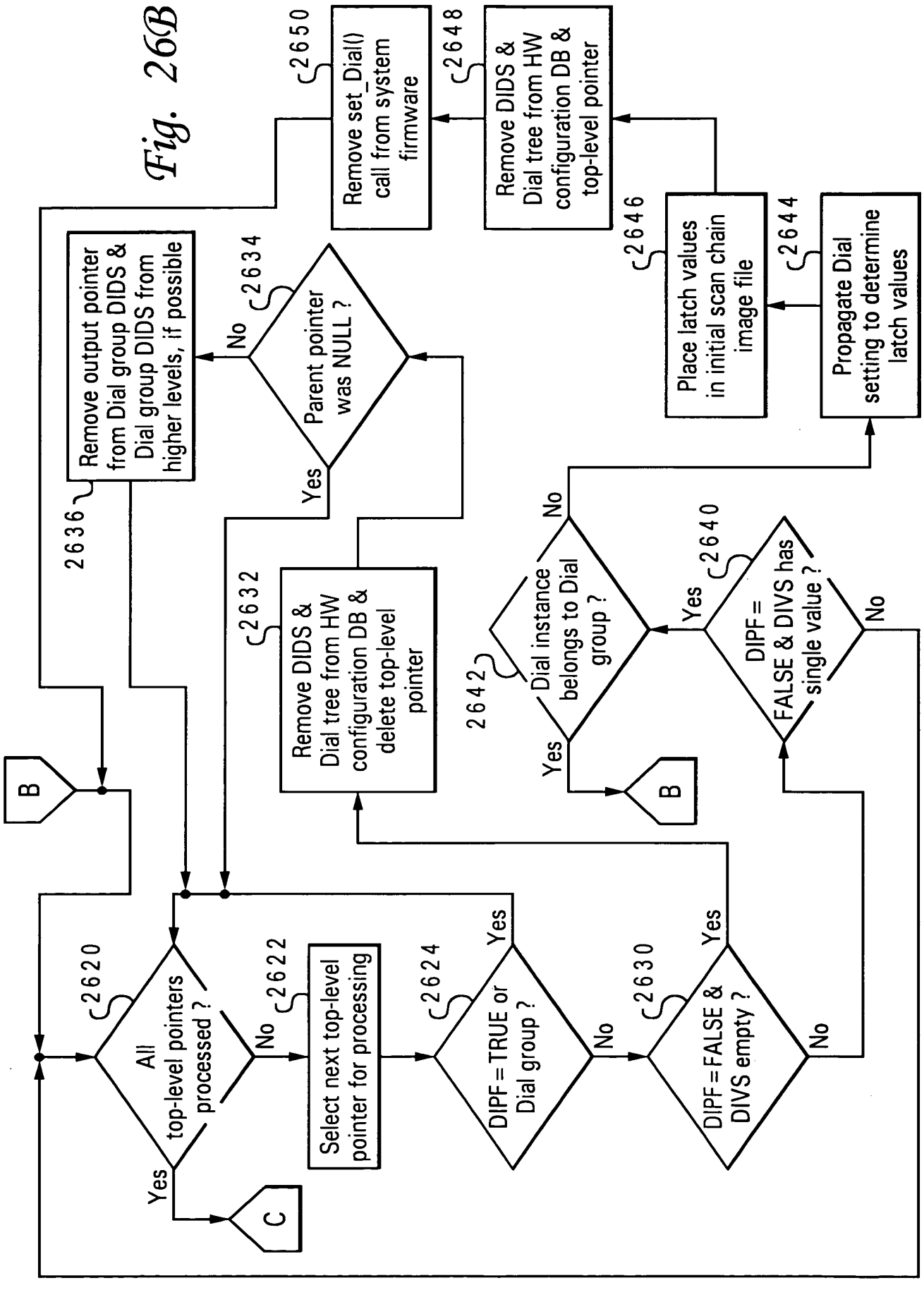
B
2620
All top-level pointers processed ?
Yes
C
No
2622
Select next top-level pointer for processing
2624
DIPF = TRUE or Dial group ?
Yes
No
2630
DIPF = FALSE & DIVS empty ?
Yes
No
2640
DIPF = FALSE & DIVS has single value ?
Yes
No
2642
Dial instance belongs to Dial group ?
Yes
No
B
2632
Remove DIDS & Dial tree from HW configuration DB & delete top-level pointer
2634
Parent pointer was NULL ?
Yes
No
2636
Remove output pointer from Dial group DIDS & Dial group DIDS from higher levels, if possible
2644
Propagate Dial setting to determine latch values
2646
Place latch values in initial scan chain image file
2648
Remove DIDS & Dial tree from HW configuration DB & top-level pointer
2650
Remove set_Dial() call from system firmware

Fig. 26B

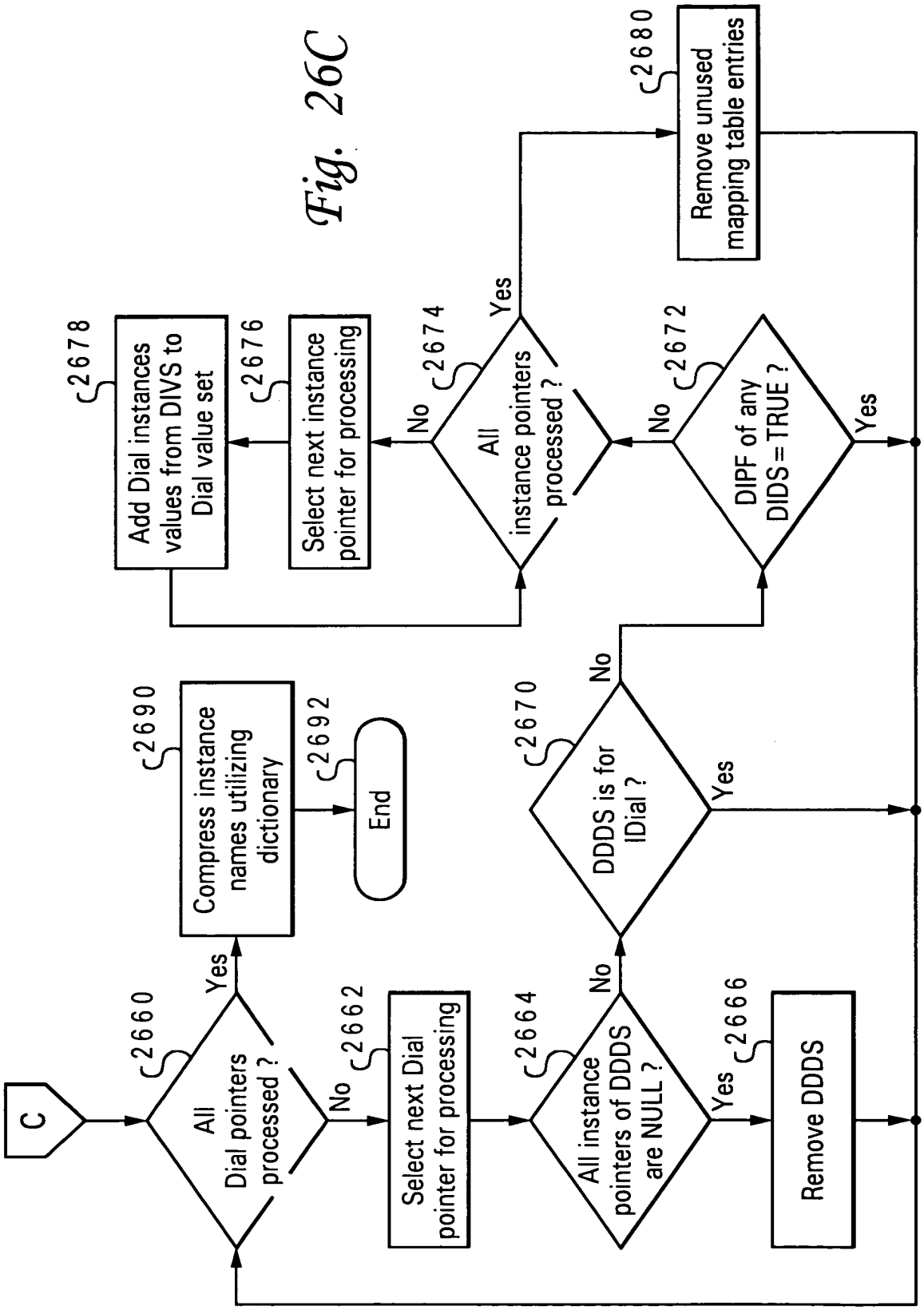
C
2660
All Dial pointers processed ?
Yes
No
2662
Select next Dial pointer for processing
2664
All instance pointers of DDDS are NULL ?
No
Yes
2666
Remove DDDS
2670
DDDS is for IDial ?
No
Yes
2672
DIPF of any DIDS = TRUE ?
No
Yes
2674
All instance pointers processed ?
No
Yes
2676
Select next instance pointer for processing
2678
Add Dial instances values from DIVS to Dial value set
2680
Remove unused mapping table entries
2690
Compress instance names utilizing dictionary
2692
End

Fig. 26C

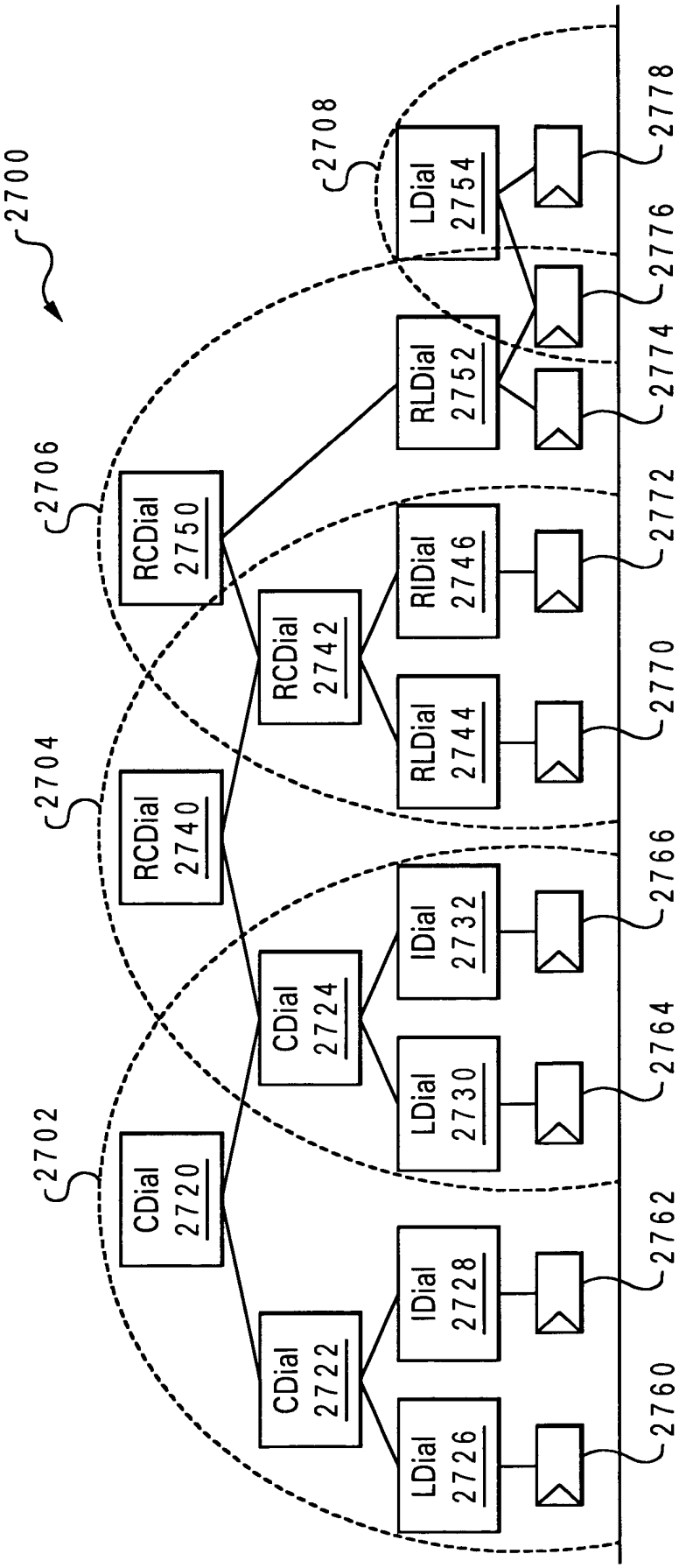
2700
2702
2704
2706
2708
CDial 2720
CDial 2722
CDial 2724
RCDial 2740
RCDial 2742
RCDial 2750
RLDial 2752
LDial 2754
LDial 2726
IDial 2728
LDial 2730
IDial 2732
RLDial 2744
RIDial 2746
2760
2762
2764
2766
2770
2772
2774
2776
2778

Fig. 27

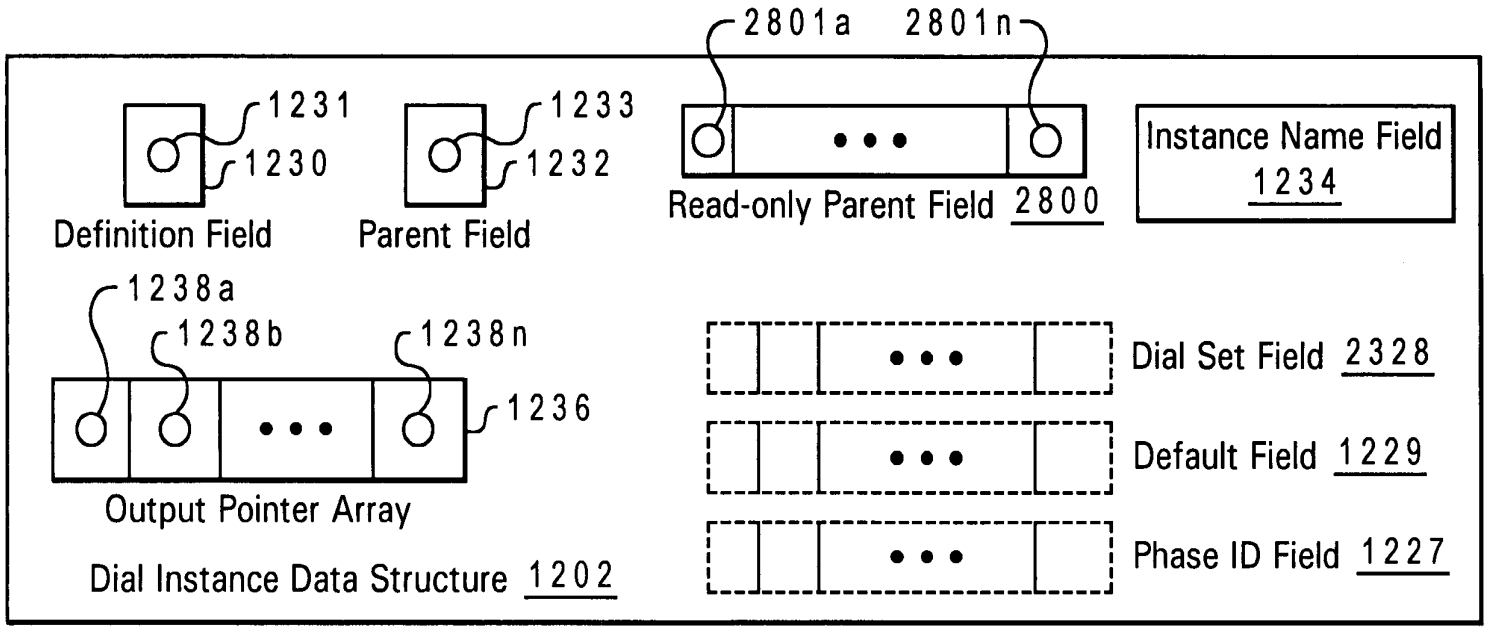

Fig. 28A

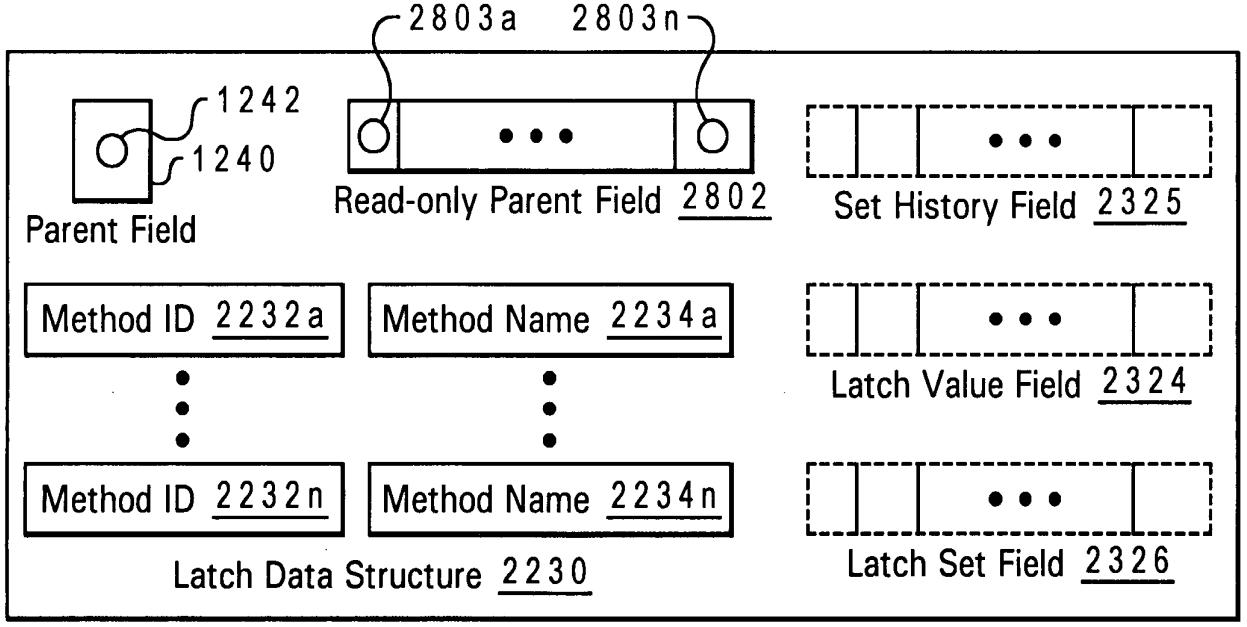

Fig. 28B

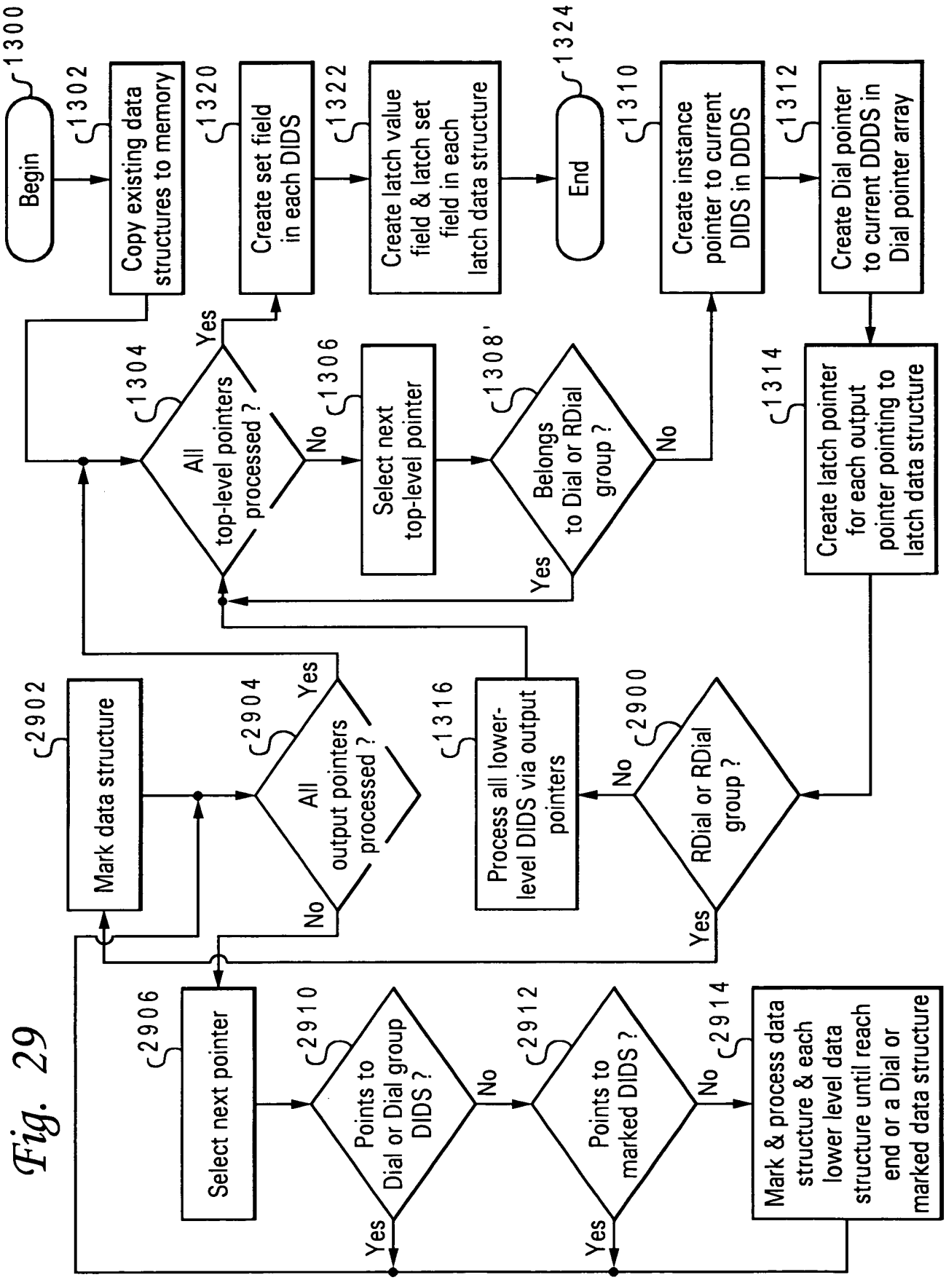
1300
Begin
1302
Copy existing data structures to memory
1320
Create set field in each DIDS
1322
Create latch value field & latch set field in each latch data structure
1324
End
1310
Create instance pointer to current DIDS in DDDS
1312
Create Dial pointer to current DDDS in Dial pointer array
1314
Create latch pointer for each output pointer pointing to latch data structure
1304
All top-level pointers processed ?
Yes
No
1306
Select next top-level pointer
1308'
Belongs to Dial or RDial group ?
Yes
No
2902
Mark data structure
2904
All output pointers processed ?
Yes
No
1316
Process all lower-level DIDS via output pointers
2900
RDial or RDial group ?
No
Yes
2906
Select next pointer
2910
Points to Dial or Dial group DIDS ?
Yes
No
2912
Points to marked DIDS ?
Yes
No
2914
Mark & process data structure & each lower level data structure until reach end or a Dial or marked data structure

Fig. 29

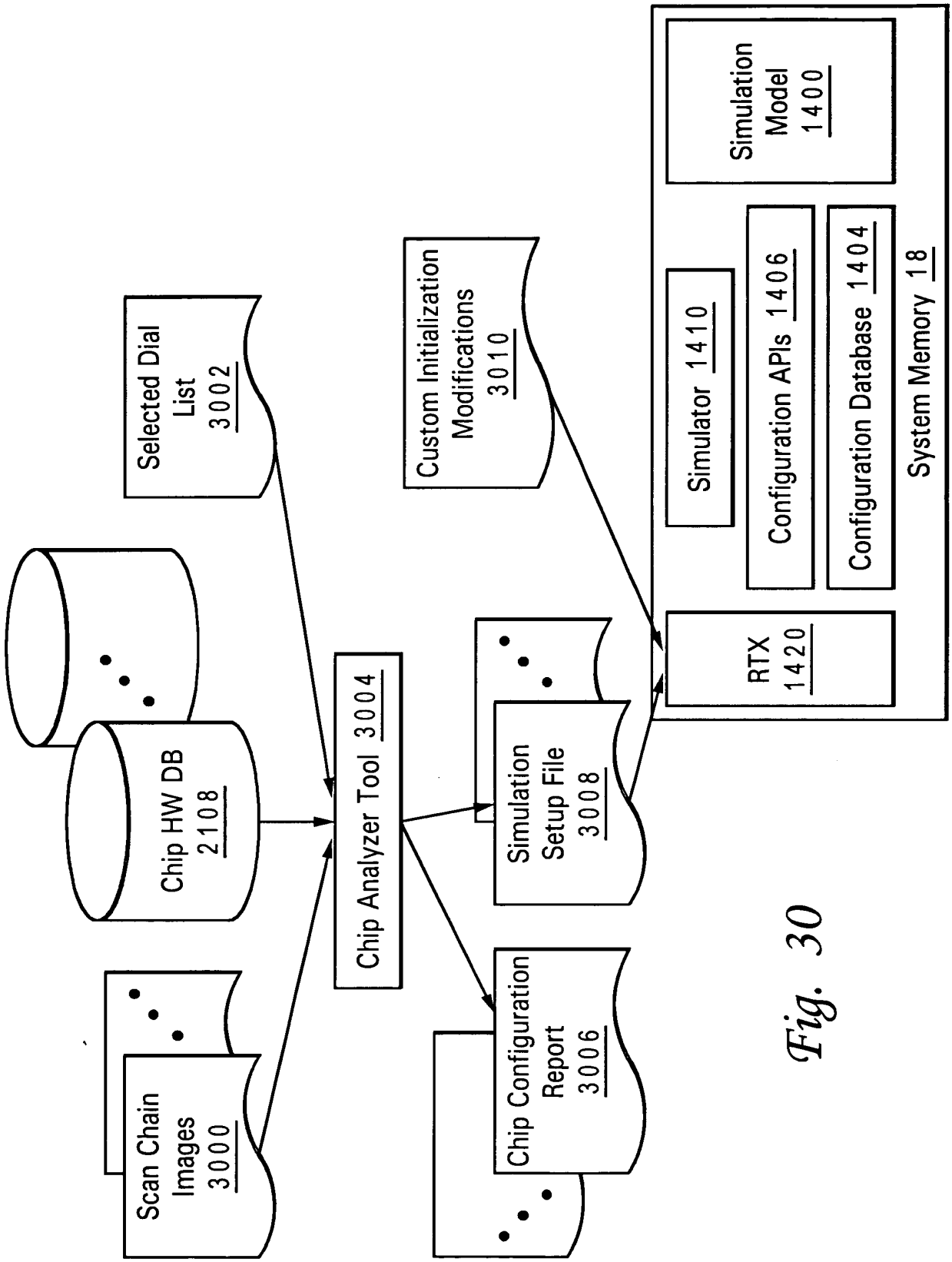
Selected Dial List 3002
Custom Initialization Modifications 3010
Chip HW DB 2108
Chip Analyzer Tool 3004
Simulation Setup File 3008
Scan Chain Images 3000
Chip Configuration Report 3006
Simulation Model 1400
Simulator 1410
Configuration APIs 1406
Configuration Database 1404
RTX 1420
System Memory 18

Fig. 30

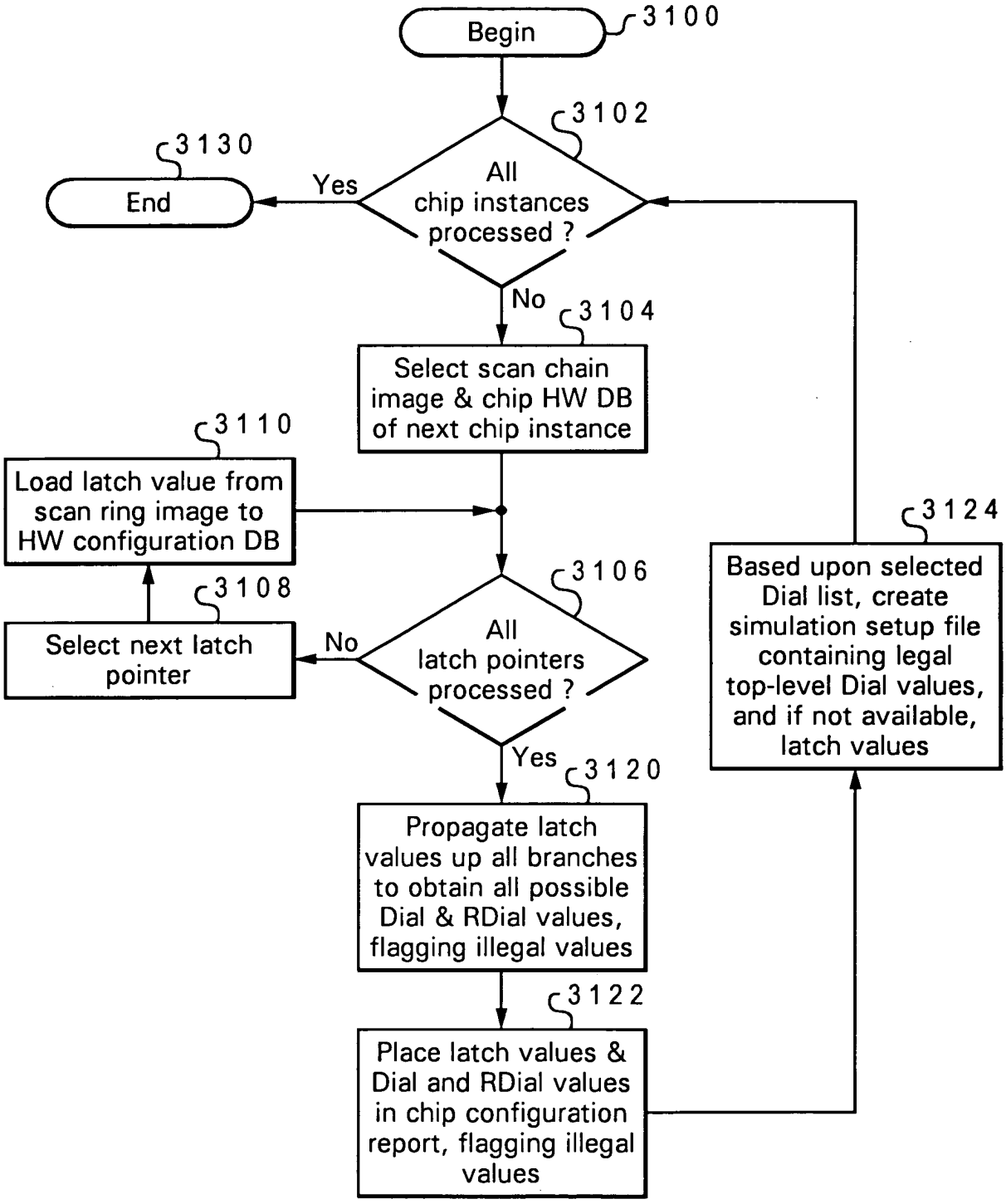
Begin
3100
3102
All chip instances processed ?
Yes
3130
End
No
3104
Select scan chain image & chip HW DB of next chip instance
3110
Load latch value from scan ring image to HW configuration DB
3108
Select next latch pointer
3106
All latch pointers processed ?
No
Yes
3120
Propagate latch values up all branches to obtain all possible Dial & RDial values, flagging illegal values
3122
Place latch values & Dial and RDial values in chip configuration report, flagging illegal values
3124
Based upon selected Dial list, create simulation setup file containing legal top-level Dial values, and if not available, latch values

Fig. 31